Efficient fall activity recognition by combining shape and motion features

Abderrazak Iazzi¹ (✉), Mohammed Rziza¹, and Rachid Oulad Haj Thami²

© The Author(s) 2020.

Abstract This paper presents a vision-based system for recognizing when elderly adults fall. A fall is characterized by shape deformation and high motion. We represent shape variation using three features, the aspect ratio of the bounding box, the orientation of an ellipse representing the body, and the aspect ratio of the projection histogram. For motion variation, we extract several features from three blocks corresponding to the head, center of the body, and feet using optical flow. For each block, we compute the speed and the direction of motion. Each activity is represented by a feature vector constructed from variations in shape and motion features for a set of frames. A support vector machine is used to classify fall and non-fall activities. Experiments on three different datasets show the effectiveness of our proposed method.

Keywords fall detection; elderly people; shape features; motion features; classification

1 Introduction

According to a report [1], each year, millions of elderly people (65 years old and over) fall. More than one in four elderly people fall each year, but less than half tell their doctors. Falling once doubles an elderly adult's chances of falling again. Falls are the leading cause of injury in adults aged 65 or older. Falls can also have an impact on the person both economically and psychologically [2]. A serious fall

can result in decreased functional independence and quality of life. Fear of falling and loss of mobility and independence are frequent and often serious consequences of a fall [2]. The risk of falling increases with age for many reasons, including overall weakness and frailty, balance problems, cognitive problems, vision problems, medication, acute illness, and other environmental hazards. As most elderly adults live alone at home or in nursing homes [3], falls can prove fatal if the elderly person does not get timely assistance. Effective detection and prevention of falls could substantially reduce disability among the elderly. Hence, there is an urgent need to develop an efficient fall detection and prevention system for monitoring elderly people. When an elderly person falls, a fall detection system can send an alarm signal to caregivers (e.g., hospitals, health centers, and family members). Such systems have recently become a significant research topic for many scientists worldwide, and many systems have been proposed for fall detection and fall prevention to help elderly people live in a secure environment with a good quality of life.

In recent years, several approaches and algorithms have been proposed for fall detection, and the number of papers on this topic has increased rapidly. Overviews of this topic can be found in Refs. [4, 5]. A recent survey [2] presented a taxonomy of fall detection from the perspective of the available fall data. Fall detection systems may be divided into three categories, based on wearable sensors [6–10], computer-vision [11–16], and ambient systems [17]. Wearable sensors (e.g., accelerometers, gyroscopes, help buttons, and acoustic sensors) capture information such as human body movements or orientation, or sound; this information is used to determine a fall. But such fall detectors have some

1 LRIT, RABAT IT CENTER, Faculty of Sciences, University of Mohammed 5 in Rabat, Morocco. E-mail: A. Iazzi, abderrazak.iazzi@gmail.com (✉); M. Rziza, mohammed.rziza@gmail.com.

2 ADMIR LAB, IRDA, RABAT IT CENTER, ENSIAS, University of Mohammed 5 in Rabat, Morocco. E-mail: rachid.ouladhajthami@gmail.com.

Manuscript received: 2020-02-16; accepted: 2020-06-08

weaknesses, e.g., the elderly do not feel comfortable wearing them, and they can forget to put them on. Also, such sensors need to be charged periodically. Computer vision can be a more suitable approach to solving this problem, as the human has no direct interaction with the camera or the system. A fall is detected automatically by video analysis. The last category, ambient systems, combine wearable sensors and computer vision.

Many approaches have been proposed for fall detection based on analysing normal and abnormal activities with computer vision, and a few datasets are publicly available for testing. These approaches can be classified as thresholding based [12, 13, 18] or machine learning based [11, 14, 15, 19–24]. In this paper, we propose a new fall recognition system based on machine learning using a single camera. Shape and motion features are extracted and combined for classification.

The organization of this paper is as follows: in Section 2, we present methods related to visual-based fall detection techniques; details of the proposed approach are given in Section 3. Then, we describe datasets, experimental results, and performance of the proposed scheme in Section 4. In Section 5, we discuss the proposed system. Finally, we give a general conclusion and we discuss future work in Section 6.

2 Related work

Because fall detection based on computer vision is a key approach for helping elderly people to live in a secure environment, there are many articles focused on building a powerful system with few false alarms and good detection rate. Some conventional techniques for fall detection are based on rules for detecting a fall. In Ref. [25], a fall is detected if the aspect ratio of the box bounding for a human exceeds a threshold, while in Ref. [18], the authors detect an abnormal event when motion exceeds a threshold, followed by no further movement during a group of frames.

However, these methods have to be adapted following any change in the position of the camera or the environment. Thus, researchers have proposed methods based on machine learning to realize a more general system. Support vector machines (SVM), convolutional neural networks (CNN), extreme

learning machines (ELM), Gaussian mixture models (GMM), naive Bayes (NB), and k -nearest neighbors (KNN) are the most common machine learning techniques used in Refs. [11, 14–16, 19–23].

In Ref. [14], the authors exploit RGB and depth information provided by a depth camera to extract several shape and motion features, both 2D and 3D, from the human silhouette. They concatenated these features into a single vector, which is fed into an SVM to classify the activity. In Ref. [11], Fisher vector encoding is used to describe the actions based on curvature scale space. A pre-trained SVM classifier is employed for the final classification. The human shape is used also in Ref. [16] for fall detection. The authors defined 5 occupancy regions. These regions were obtained by simple partitioning centered on the body's center of mass. The area ratios for each frame are calculated and used as input data for fall detection and classification. To improve the result, the authors combined the generalized likelihood ratio (GLR) and an SVM. In Ref. [23], a fall event is detected based on shape analysis by using silhouette orientation volume (SOV) features constructed from a spatio-temporal silhouette orientation image (SOI) [26]. First, each human action is represented by a bag-of-words (BoW) of the SOV. Then, an NB classifier is used to separate fall actions from normal activities. The weakness of these methods [11, 16, 26] is that success can decrease if there is occlusion or incorrect segmentation of the human shape. Additionally, motion information is not used, yet it can help to improve the results. Ismail et al. [20] proposed an approach that aims to detect a fall using membership-based histogram descriptors (MHD) as a generalization of BoW, the descriptor is obtained by mapping the original low-level visual features to a more discriminative descriptor using probabilistic memberships. The histogram oriented gradient (HOG) feature is extracted as a low-level feature. KNN was used to assign each descriptor to either fall or non-fall class.

Another method was presented in Ref. [15] where the authors used multi-view voting of results output by a GMM classifier to detect a fall based only on shape deformation between two consecutive frames of a video. Using only shape information limits this method. A combination of appearance, shape, and motion features is used in Ref. [27] to detect a fall. Each feature is represented as a moving point on

a Riemannian manifold. The velocity statistics of this point on the manifold are used with an SVM classifier to distinguish between falls and normal activity. While their experiment results show high accuracy, the processing time is excessive. Recently, Fan et al. [21] extracted several features from an ellipse computed from a silhouette to describe human posture. They then developed an SVM model to classify human posture related to a fall in each frame. They considered fall incidents as shape feature sequences and analysed them using slow feature analysis (SFA) [28]. Six shape features were extracted from the human silhouette, and these features were transformed to slow feature sequences which in a fall can be described by the accumulation of squared first order temporal derivatives of these slow features. A directed acyclic graph SVM (DAG-SVM) is used to detect falls.

CNNs have had great success in the field of pattern

recognition. A brief introduction can be found in Ref. [29]. Effective features are extracted using a CNN to perform image detection and classification. The authors in Ref. [22] proposed a vision-based solution using a CNN to detect if a video sequence contains fall incidents. The solution takes an optical flow image as input to incorporate motion information.

3 Proposed approach

Analyzing human shape and motion variation are the most common approach for human behavior recognition such as fall detection. A fall is characterized by shape deformation and rapid motion. However, some normal activities are similar to a fall. To overcome this problem, we propose a novel approach that includes three phases, human shape extraction, feature extraction with segmentation, and fall detection through classification. We start with human shape extraction from video input using background subtraction, followed by post-processing to extract the human silhouette, and then update the background model. Then, a bounding box is fitted to the human silhouette to extract several features that accurately describe the current human posture. Initially, we represent the shape of the person with three blocks based on the bounding box of the human silhouette, as shown in Fig. 2. We divide the bounding box into three blocks. Then, we compute the velocity inside each block and the person's velocity in order to analyze the motion variation. Also, we use the bounding box, an ellipse around the human shape, and projection histogram features to analyze shape variation. We extract motion and shape features from a group of frames, and concatenate them into a single feature vector to represent the activity. Finally, this vector is fed into an SVM classifier to classify the activity. A fall is confirmed using the floor region and a strategy of majority voting.

3.1 Human detection and floor segmentation

3.1.1 Background subtraction

The first step in our proposed method is extracting the human silhouette from the background. There are many proposed algorithms in the literature for moving object detection such as a GMM-based algorithm [30], a codebook model (CB) [31], and approximated median filters (AMF) [32]. For a comparison and

Fig. 1 Block diagram of our approach.

Fig. 2 Our three block representation for human body shape and motion.

further details, see Ref. [33]. Another approach using signal decomposition was proposed in Ref. [34]. This approach is unsuitable for this problem as it takes too much time for image processing. Deep learning has also been applied for background subtraction, e.g., in Ref. [35]. While it has advantages for extracting the human silhouette, the processing time and need for training datasets are drawbacks.

The result of background subtraction (BS) is not always satisfying because of shadows and moving furniture in the background. For our system, we thus used a CB algorithm due to its advantages and robustness when detecting moving objects and its ability to remove shadows. We initially detect and remove shadows from the foreground with the method in Ref. [36]. We determine shadows using HSV color and gradient information, and then classify shadow pixels based on pre-defined thresholds. The result obtained contains many objects. In order to determine the human silhouette, we use two rules: (i) remove all blobs with a small area (< 50 pixels) and (ii) blob merging to place the blobs in classes using the rectangle distance, defined as the minimum 4-distance [37] between two rectangles:

$$D(B_1, B_2) = \min_{P_1 \in R_1, P_2 \in R_2} d_4(P_1, P_2) \quad (1)$$

where B_1 , B_2 correspond to blob 1 and blob 2, and P_1 and P_2 are the closest points for rectangles R_1 and R_2 respectively. If the distance between two blobs is less than 50 pixels, then they are in the same class; otherwise, each is in different class. Figure 3

Fig. 3 Four examples of distances between two blobs.

shows the distances between various pairs of blobs in different positions.

Then, we determine the human silhouette by using the motion of the blob's pixels based on optical flow and the distance between the current and previous human position for each class. Blobs with a small distance and large motion are taken as the required blobs.

Figure 4 illustrates our human silhouette extraction method. First, we apply a CB method to detect moving objects in the background (c), and then, we detected shadows (d). Next, by simple subtraction, we removed the shadows from the CB result (e). Finally, we removed the small areas (blobs) using blob merging to give the last frame (f).

3.1.2 Floor extraction

As a fall event always ends on the ground, the floor must first be detected, to allow us to confirm whether

Fig. 4 Extracting a human from the background: (a) reference frame, (b) current frame, (c) CB result, (d) shadow detection, (e) difference between (c) and (d), (f) segmented human silhouette.

the current activity can be classified as a fall or normal activity. Many unsupervised methods [38, 39] have been proposed to find the floor plane in a frame. In Ref. [39], the authors mark the ground as the region near the lower extreme point of the human ellipse when the person is classified as standing or sitting. Alternatively, the authors in Ref. [38] estimate the floor plane by using the disparity map from a Kinect camera. Supervised methods have also been proposed—see the survey in Ref. [40]. In this paper, we considered applying the work in Ref. [41], which uses a SegNet model, a deep encoder–decoder architecture for multi-class pixel segmentation. It is effective for indoor scene understanding and also for road scene segmentation. It can segment 37 indoor scene classes, including wall, floor, ceiling, table, chair, sofa, and others. However, humans are not considered. Thus, the only way to use this method in fall detection is in the initial frame of a video, which does not include a person. Figure 5 shows a result obtained using this method on our dataset; the floor corresponds to the green pixels. It gives good results when the person is not inside the frame.

However, we base our approach on using a manual process for floor extraction, for several reasons: the SegNet method does not support human segmentation, and there is no depth information in our dataset. Most recorded dataset videos are short, which leads to an inability to construct a floor segment from human postures as in Refs. [38, 39]. Considering these limits in our datasets, we decided to extract floor information manually to demonstrate

Fig. 5 Floor extraction using the SegNet method. Green pixels correspond to the floor.

its effectiveness as a feature in our fall detection system.

3.2 Feature extraction

After background subtraction, we use the human silhouette to extract several features based on the bounding box and ellipse. Both the motion and shape information play a significant role in detecting and analyzing human activity. A fall is characterized by shape deformation and rapid motion. We define several features, given in Table 1, to describe shape and motion variation.

3.2.1 Shape variation

To describe shape deformation, we extract the three features F_5 , F_6 , and F_7 .

Feature F_5 can be computed from the bounding box (BB) drawn around the person as shown in Fig. 6. The *aspect ratio of the BB* (R_B) is the ratio of its height to its width. During a person's activities at home, the height and width will change as the person changes posture, so R_B will change too.

Table 1 Features

Feature	Description
F_1	Person's velocity, based on optical flow
F_2	Head block velocity, based on optical flow
F_3	Center block velocity, based on optical flow
F_4	Feet block velocity, based on optical flow
F_5	Aspect ratio of bounding box
F_6	Aspect ratio of projection histogram
F_7	Ellipse orientation
F_8	Motion direction

Fig. 6 Human silhouette extraction and three block representation. The word “left” in (c) indicates the person's direction.

Feature F_6 is extracted from the projection histogram based on the person's 2D silhouette. See Fig. 7. We project the silhouette onto the x - and y -axes. Each histogram bin corresponds to the number of white pixels in one row or column of the image. The highest values in horizontal and vertical bins are H_p and V_p . The *histogram aspect ratio* (R_H) of H_p to V_p is taken as a feature.

Feature F_7 can be extracted from the ellipse fitted to the human body as shown in Fig. 6. A moment-based method [33] is applied to fit the ellipse. From this, we extract the person's *orientation*, θ . See Fig. 2 for an example of the fitted ellipse.

As each activity takes several frames, we analyze shape deformation over a window of W frames. Variation of shape deformation within it can be represented by a feature vector defined as follows:

$$\mathbf{F}_s = [\mathbf{V}_{R_B}, \mathbf{V}_{R_H}, \mathbf{V}_\theta] \quad (2)$$

where the vector $\mathbf{V}_{R_B}$, $\mathbf{V}_{R_H}$, and $\mathbf{V}_\theta$ represent vectors of features F_5 , F_6 , and F_7 over W frames respectively.

$$\begin{aligned} \mathbf{V}_{R_B} &= [R_b^1, \dots, R_b^W] \\ \mathbf{V}_{R_H} &= [R_h^1, \dots, R_h^W] \\ \mathbf{V}_\theta &= [\theta^1, \dots, \theta^W] \end{aligned} \quad (3)$$

3.2.2 Motion estimation

Optical flow is a visual displacement field that helps to explain variations in a moving image in terms of displacement of image points. There are several approaches to motion detection using optical flow, e.g., Refs. [42, 43], where the authors calculate the flow for each pixel in the first image, and then use multi-scale tracking of sparse features.

We used the algorithm in Ref. [42], as tracking with image pyramids allows large motions to be caught by local windows. Optical flow can give two

important pieces of information for analyzing human behavior: the person's *speed* and their *direction* of motion. These allow discrimination between normal and abnormal activities.

Some normal activities are similar to a fall event. Using only the person's speed is not enough to distinguish between these activities. During a fall, compared to lying down, the head and center of the body move quickly while the feet remain almost still. Hence, we represent the person's shape using three blocks by dividing the BB into head, center, and feet blocks as shown in Fig. 2. The three blocks have sizes in the ratio 1:2:2, based on experiments.

From the person's blob and the three blocks we compute the person's velocity v_p (feature F_1) and the velocity of each block v_i , $i = 1, 2, 3$ (features F_2 – F_4) from the optical flow results as follows:

$$v_i = \frac{\sum_{\text{Pixels in person blob} \cap B_i} v}{N_i} \quad (4)$$

$$v_p = \frac{1}{3} \sum_{i=1}^3 v_i \quad (5)$$

where v is the velocity of a pixel and N_i is the number of pixels in block B_i .

The motion direction (F_8) is the last feature extracted from the optical flow result. The orientation of the pixel's displacement is used to compute the motion direction. Four directions are defined, namely *up*, *left*, *right*, and *down* as shown in Fig. 8. We give the down (fall) direction a value of 1 and the other directions a value of 0.

Since a fall, like other activities, is characterized by variation in motion, it is necessary to capture this information from a group of frames. We extract five

Fig. 7 Human silhouette projection histogram.

Fig. 8 Directions of motion.

feature vectors of motion from W frames which are defined as follows:

$$\mathbf{V}_P = [v_p^1, \dots, v_p^W] \quad (6)$$

$$\mathbf{V}_H = [v_1^1, \dots, v_1^W] \quad (7)$$

$$\mathbf{V}_C = [v_2^1, \dots, v_2^W] \quad (8)$$

$$\mathbf{V}_F = [v_3^1, \dots, v_3^W] \quad (9)$$

$$\mathbf{V}_D = [d^1, \dots, d^W] \quad (10)$$

where $\mathbf{V}_P$ is the vector for the person's speeds, $\mathbf{V}_H$, $\mathbf{V}_C$, and $\mathbf{V}_F$ are vectors of the head's, center's, and feet's speeds, and $\mathbf{V}_D$ is the vector of motion directions.

Finally, we concatenated all these vectors to construct a motion feature vector as follows:

$$\mathbf{F}_m = [\mathbf{V}_P, \mathbf{V}_H, \mathbf{V}_C, \mathbf{V}_F, \mathbf{V}_D] \quad (11)$$

3.3 Feature vector

Each video segment is characterized by a feature vector which concatenates motion features and shape features as follows:

$$\mathbf{V} = [\mathbf{F}_m, \mathbf{F}_s] \quad (12)$$

This vector is fed into a classifier. We tested our approach using various classifiers such as an SVM, a neural network (NN), and an ELM.

3.4 Fall activity classification

Fall detection is formulated as a binary classification problem that distinguishes falls from daily living activities (DLAs), so the total number of classes $K = 2$. All normal activities (walking, sitting, lying down, squatting, and bending) are treated as a single negative class. Walking is included as it is the most common activity in daily life. The others are considered because they have similarities to falling, so might confuse the classifier.

Let the training set be $X = (x_i, y_i)$, $i = 1, \dots, N$, where x_i is the feature vector for some activity, y_i is its class label ($y_i \in \{-1, 1\}$) and N is the number of training data instances. A binary SVM classifier [44] is trained with X . Presenting a feature vector x representing a test activity, the SVM classifier returns the signed output margin a , where $y = \text{sign}(a)$ is the determined class label for input x . A fall is indicated by $y = +1$.

Figure 9 shows the general scheme of fall and normal activity classification. For each activity, we extracted the features and concatenated into a single feature vector. N example activities including falls and DLAs are used to construct the training data for

the SVM classifier, which is then used to classify new activities as falls or DLAs in the testing phase.

3.5 Video segmentation

We now describe our manual method for obtaining a video segment that contains only an event of interest (i.e., an activity) from a video. Segmentation is based on fall sequence duration. Volunteers of different ages and height simulated the activities. Also, videos were captured at different frame rates (25 fps, 30 fps, and 120 fps). As a result, the duration of activities varies between datasets. By considering all videos, we observed that the length of a fall sequence is typically 20 or 40 frames, starting from standing or sitting and ending with lying on the ground. So that all video segments result in feature vectors of the same length, we decided to divide the activity sequence of 40 frames into two segments of 20 frames, composed of the odd-numbered and even-numbered frames respectively. Thus, we took 20 frames to be the duration of any activity sequence in all experiments. Another factor we considered was the frame size: the databases used contained video from different types of cameras, so we normalized all frame sizes to 320×240 .

3.6 Video fall detection

Given a video, we need to determine the beginning and the end of the activity for classification. As noted in the previous section, human activity is segmented manually. There are many techniques that automatically perform video segmentation of human activity, based on a hidden Markov model (HMM) [45] or Markov chain Monte Carlo (MCMC) [46] methods. In Ref. [27], the authors observed that the aspect ratio of the BB is relatively low when the person is walking or standing, but the ratio increases when a person falls. Thus, segmentation is performed by finding the place in the video when there is a significant increase in the target's aspect ratio. They used MCMC to find this transition.

Our approach is based on a strategy of majority voting. We have no information about the beginning of the activity. Furthermore, between the start and end there are many similar activities, so we use a temporal window as shown in Fig. 10. Let T be the duration of the temporal window, and $\mathbf{A} = \{a_1, \dots, a_T\}$ be the set of classified activities in this window, where $a_j \in Y = \{-1, 1\}$, Y being the classification. If there are more than ten occurrences

Fig. 9 Scheme for recognizing human activities.

of activity classified as a fall within a temporal window, then we considered this window as a whole to correspond to a fall incident.

We used floor information and inactivity time to confirm the fall. We check if the body is mainly inside the ground region, as shown in Fig. 11. A fall is confirmed if more than 75% of the body is inside the ground and inactivity lasts for more than ten frames.

Fig. 11 A fallen person lies on the floor. Blue region: detected floor. Red region: intersection of the foreground human body with the floor.

4 Experiments, evaluation, and results

4.1 Datasets

To evaluate our proposed method, we used three datasets [12, 47, 48]. Figure 12 shows some key frames of video segments from these datasets. Tables 2 and 3 give quantitative information about, and some challenges presented by, the different datasets. All videos contain only one person, as our system is not needed if there is more than one person present: the other person could help fallen person. More details about each dataset are provided below.

Fig. 10 Fall detection in video.

Fig. 12 Key frames from several datasets showing falls and normal activities. Rows 1–3: fall activities from datasets D_1 – D_3 respectively. Rows 4, 5: sitting and walking activities from dataset D_2 .

Table 2 Difficulties and challenges of each dataset

Dataset	Activities	View angle	Background	Texture	Shadows	Furniture	Occlusion
D_1	all types	various	change, simple	difficult	yes	yes	no
D_2	all types	various	fixed, simple	difficult	yes	no	no
D_3	all types	various	change, cluttered	more difficult	yes	yes	yes

Table 3 Number of fall and DLA activities in each dataset

Class	Activity	D_1	D_2	D_3
+1	Fall	29	234	200
–1	DLA	45	356	288

D_1 [12]: This dataset was captured in a room using a single RGB camera positioned on the ceiling. Different volunteers simulated several activities such as falling, walking, sitting, lying, crouching, and

bending. It contains 20 videos with 29 falls and 45 normal activities; each video may contain more than one activities. We segmented each activity using the method in Section 3.5.

D_2 [47]: This dataset was recorded in multiple rooms (coffee room, lecture room, home, and office) with a single RGB camera positioned 2 meters above the floor. Different volunteers simulated several activities such as falling, walking, sitting, and other

activities. The dataset contains 249 videos with 192 showing a fall incident and 57 showing normal activities. Each activity's duration is on average 40 frames. Thus, we created new datasets, $D_{2.1}$ & $D_{2.2}$ from this dataset using the method in Section 3.5. Each scenario, $D_{2.1}$ or $D_{2.2}$ contains 117 segments with falls and 178 with normal activities, giving a total of 234 fall segments and 356 DLA segments.

D_3 [48]: This dataset was created using 8 calibrated RGB cameras to record 24 scenarios. One volunteer simulated all activities. As the videos in each scenario were recorded from different views, we mixed all videos in our experiments. We segmented falls and many other normal activities from each video using the method in Section 3.5. This resulted in 200 fall segments and 288 normal activity segments.

4.2 Classification and data analysis

In our experiment, feature extraction from the videos was implemented in C/C++ using the OpenCV Library with Visual Studio 2013. The classification procedure was implemented in MATLAB. All experiments were conducted on a PC with an Intel core i7 CPU and 12 GB RAM. Based on libSVM [44], we used the C-SVM classifier with radial basis functions (RBF). Default parameters were used except for gamma in the RBF, set to $g = 0.01$ and cost in the C-SVM, set to $c = 100$. 10-fold cross-validation was used. Before presenting the performance of our system, we illustrate results of the BS method and features extracted from the activities.

Figures 13–15 show three activities (walking, falling, and lying down). Note the differences in the graphs, especially in motion features and variation of orientation. The trajectories for walking activity are fairly stable, while in the other activities, there are sudden changes. The difference between falling and lying down is that the variations in features occur slowly for lying down, but rapidly for a fall.

4.3 Performance criteria

To evaluate the effectiveness of our features, and the proposed method, we used several indicators based on true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). Precision and accuracy indicators are defined as follow:

$$\text{Precision} = \frac{TP}{TP + FN} \quad (13)$$

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (14)$$

Fig. 13 Walking activity. Row 1: input frames. Row 2: extracted human silhouettes. Row 3: three blocks representation. Graph: variation in feature values over 20 frames.

Fig. 14 Falling activity. Row 1: input frames. Row 2: extracted human silhouettes. Row 3: three blocks representation. Graph: variation in feature values over 20 frames.

4.4 Feature evaluation

In this section, we evaluate classification using shape features, motion features, and their combination. The SVM classifier was used with 10-fold validation.

Table 4 shows the classification results using each kind of feature. It is clear that using combined features provides high rate of accuracy than using either shape or motion features alone.

Fig. 15 Lying down activity. Row 1: input frames. Row 2: extracted human silhouettes. Row 3: three blocks representation. Graph: variation in feature values over 20 frames.

Table 4 Classification score for different kinds of feature

Dataset	Shape	Motion	Both
D_1	93.24%	89.18%	94.60%
$D_{2.1}$	89.15%	94.91%	95.93%
$D_{2.2}$	90.81%	94.89%	95.91%
D_3	90.36%	91.59%	96.31%

4.5 Testing, performance, and comparison

We now test our proposed approach in two ways. The first applies our approach to each database individually, while the second uses a combination of databases.

Table 5 gives the accuracy achieved by the proposed method for individual datasets. The proposed scheme shows high accuracy for databases D_2 and D_3 , but lower for D_1 , as its training phase has insufficient data to discriminate between similar activities. Another possible issue is that the dataset contains some activities that are extremely similar to a fall, for example, crouching and lying down activities have large motions and shape deformations.

When testing with multiple databases, Table 6 shows that results obtained from the proposed method are improved in some cases and less good in others. When we merged datasets $D_{2.1}$ and $D_{2.2}$, the accuracy was 98.12%: the system learns well from this large dataset and can effectively discriminate

Table 5 Accuracy of the proposed fall detection system for individual datasets

Dataset	Accuracy
D_1	94.60%
$D_{2.1}$	95.91%
$D_{2.2}$	95.23%
D_3	96.31%

Table 6 Accuracy of the proposed fall detection system for combined datasets

Dataset	Accuracy
$D_1 + D_{2.1}$	95.65%
$D_1 + D_{2.2}$	94.83%
$D_1 + D_{2.1} + D_{2.2}$	97.43%
$D_1 + D_3$	94.66%
$D_{2.1} + D_{2.2}$	98.12%
$D_{2.1} + D_{2.2} + D_3$	96.75%
$D_1 + D_{2.1} + D_{2.2} + D_3$	96%

between similar activities. When we further merged D_1 with $D_{2.1}$ and $D_{2.2}$, the accuracy went up to 97.43%. However, while merging D_3 with D_1 , or D_3 with $D_{2.2}$ and $D_{2.1}$, the results were not so good. This can be explained in that the characteristics of dataset D_3 are more complex than those of the other datasets. Another further cause may be that taking only 20 frames as the duration of each activity is insufficient to discriminate between similar activities. In the last experiment, we merged all datasets, and even though the system was able to learn from the largest amount of data in any test, the result was not the best, achieving 95.82% which is lower than the result from D_1 with $D_{2.1}$ and $D_{2.2}$.

In Table 7, we compare results using three different classifiers: SVM, NN, and ELM. As it can be seen, the SVM classifier always provides higher accuracy than other classifiers.

4.6 Comparison with related work and discussion

We now compare our proposed scheme with the methods discussed in Section 2. Our comparison is based on the performance achieved by each approach using the same dataset.

From Table 8, we observe that our proposed scheme result is better than the methods presented in Refs. [12, 15, 16, 20–22]. Methods in Refs. [12, 15, 16] are based only on shape features, and motion features are necessary to discriminate true falls and activities like a fall. The shape features of these methods gives

Table 7 Accuracy provided by different classifiers

Dataset	SVM	NN	ELM
D_1	94.60%	85%	85%
$D_{2.1}$	95.91%	96%	94.60%
$D_{2.2}$	95.23%	85.34%	85.33%
D_3	96.31%	90.98%	93.44%
$D_{2.1} + D_{2.2}$	98.12%	91.21%	93.24%
$D_1 + D_{2.1}$	95.65%	90.32%	94.62%
$D_1 + D_{2.2}$	94.83%	88.17%	88.17%
$D_1 + D_3$	94.12%	87.32%	88.73%
$D_3 + D_{2.1}$	95.01%	88.32%	87.81%
$D_3 + D_{2.2}$	94.89%	86.80%	88.32%
$D_1 + D_{2.1} + D_{2.2}$	97.43%	89.82%	94.61%
$D_3 + D_{2.1} + D_{2.2}$	96.75%	88.15%	91.11%
$D_3 + D_1 + D_{2.1} + D_{2.2}$	96%	91.35%	93.08%

Table 8 Comparison with other methods

Dataset	Method	Precision	Accuracy
D_1	Chua et al. [12]	—	91.9%
	Ours	—	94.60%
D_2	Nguyen et al. [18]	93.25%	—
	Ismail and Bchir [20]	—	94.73%
	Harrou et al. [16]	—	96.84%
	Núñez-Marcos et al. [22]	—	97.00%
	Ours	100%	98.12%
D_3	Rougier et al. [15]	—	95.6%
	Fan et al. [21]	—	94.00%
	Fan et al. [19]	—	97.10%
	Yun and Gu [27]	—	97.19%
	Ours	—	96.31%

higher accuracy than our shape features alone, but when we add motion features, our result is improved and our method achieves higher accuracy.

The proposed scheme consistently outperforms the methods in Refs. [19, 27]. We also compare our method with the work in Ref. [18]; precision is the only indicator used in that work so we use it for comparison. Our scheme shows higher performance in detecting falls with a precision of 100%, while their method achieves only 93.25%.

We also compared our method with the method in Ref. [14] which is not included in Table 8, as their system is based on an RGB-D camera. Their system achieved 97.5% accuracy, but this high performance can be explained as their system simply considered falling and lying down activities. If they were to consider other similar normal activities (e.g., crouching down, sitting rapidly), classification would be more difficult, and the accuracy would be lower.

Overall, it is difficult to compare like-with-like, as other methods used multiple cameras or merge RGB and depth information. Nevertheless, this comparison gives some indication of the performance and robustness of the proposed method.

4.7 Fall detection

Fall activity recognition and the floor extraction are used to detect fall events through the three conditions described in Section 3.6. To illustrate them, we present the four cases in Fig. 16. The first illustrates a person who falls on the floor, and the activity is classified as a fall event. As the body area is inside the area classified as ground, and there is no activity for ten frames, we confirm a fall. In the second case, we show the person is standing, and the activity is classified as a normal activity. As the largest body area is not inside the ground area, the system classifies this as normal activity. In the third case, the person's body is bending and the activity is classified as a normal activity. As the largest body area is inside the ground, our system detects this activity as normal. The last case shows lying and sitting activities. The first activity is detected as normal because the classifier considers it to be a normal activity even though the body area is inside the floor. The second one is detected as a normal activity because the classifier assigns it as a normal activity.

4.8 Processing time

For any monitoring system for fall detection, processing time plays an important part in triggering an alert quickly after the person's fall. The average time for video data pre-processing is 0.24 second/frame (s/f). The average speed of feature extraction is approximately 0.024 s/f. For fall verification, the average speed is 0.03 s/f. By considering all steps, we can see that pre-processing step takes the longest computation time, due to its use of optical flow. Further speed could be achieved by code optimization parallel computing; an FPGA could be used to produce a real time system.

Table 9 compares processing time for our system with other methods. Those in Refs. [12, 15] provide a lower execution time than our method as they only use shape information without motion information. Our method is twice as quick as the method in Ref. [27].

Fig. 16 Different activities. Row 1: falling to the floor. Row 2: standing on the floor. Row 3: sitting on a chair. Row 4: Bending. Left to right: input frame, detected human silhouette, floor (blue) and its intersection with the human body (red, white is the part of the body not intersecting with the floor), output of our system.

Table 9 Processing time for fall detection systems

Approach	Execution speed	
Chua et al. [12]	0.19 s/f	5.26 FPS
Rougier et al. [15]	0.2 s/f	5 FPS
Yun and Gu [27]	13.2 s/f	0.07 FPS
Our method	0.33 s/f	3.03 FPS

5 Discussion

The proposed approach is a fall detection system using a single camera. Computer vision provides more complete and detailed information about the supervised person (e.g., their activity, posture, and location), as well as their environment, than simpler sensors. It is convenient, as no human intervention is needed, and no sensors need be worn.

Our evaluation was conducted on three available datasets. Despite the different databases, issues in merging them, and segmentation difficulties (shadows, moving objects, different clothes, differently sized

persons, different locations, etc.), recognition works well.

Combining shape and motion features is a useful way to distinguish normal activity from abnormal activity. We have demonstrated that the shape features extracted from the human silhouette and the motion features provide good discrimination. Our procedure is totally automatic and there is no need to choose any thresholds to distinguish activities. Compared to previous methods, our approach has the following characteristics, which is its main novelties. The main features used for fall detection are shape features: the human shape is extracted via background subtraction combined with shadow detection to cope with background changes such as shadows and lighting changes. In each frame, blob operations are used to eliminate unwanted objects and add them to the background model, so that, in the next frame, they disappear from the foreground.

Combining shape and motion features improves the classification rate over using either type of feature alone, as in Refs. [12, 15, 16]. Fall confirmation is applied when abnormal activity is detected, in order to discriminate real fall activities from similar activities such as lying down and bending. To do so we use floor information and inactivity time. Most errors in previous systems are caused by this problem; confirmation using floor information reduces errors.

However, some cases are still not considered by our system; see Refs. [12, 16]. Considering multiple persons is not required. If there is more than one person present, there is no need for our system, as the other person can help if the elderly person falls. In such cases, the system can be automatically turned off. It could be restarted manually by the elderly person or automatically using some technique of counting people, e.g., see Ref. [49]. Occlusion poses a problem to our system. The home environment often contains many objects like sofas, chairs, tables, and so on. When the person is behind one of them, the resulting occlusion degrades the performance of our fall detection system. One way to avoid this problem is to add further cameras for monitoring, to guarantee that the whole body or most of it is in front of at least one camera. If each camera performs fall detection, the results can be combined using some technique such as majority voting to make a decision. Such strategy was used in Ref. [15].

6 Conclusions and future work

In this paper, we proposed a fall detection scheme for the elderly. A fall is characterized by shape deformation and high motion. We construct two feature vectors representing shape and motion variation over a group of frames. The shape features include the aspect ratio of the person's BB, the human ellipse's orientation, and the aspect ratio of the projection histogram. Motion features include the velocity of the head, center, and feet defined using three blocks, as well as the velocity and the direction of the person based on the BB. We combine these two feature vectors, into a single feature vector, and use an SVM classifier to distinguish between fall and non-fall activities. The proposed method was validated using three publicly available datasets. The results show that our fall detection scheme is more effective than other methods and has a high accuracy,

with good performance even when merging different datasets.

Despite the advantages of our method, it has limited testing on real fall accidents due to lack of publicly available data. Also problems of occlusion and cluttered backgrounds due to furniture and angle of view are still not entirely solved. Using multiple cameras is one possible approach. Further testing on more databases is planned as well as creating more scenarios to test the robustness of the proposed method. Also, we will consider separate classification of other activities instead of treating them as one class, and develop an automatic floor segmentation method.

Acknowledgements

This research is part of the ANGEL project supported by MERSFC and CNRST.

References

- [1] Bergen, G.; Stevens, M. R.; Burns, E. R. Falls and fall injuries among adults aged ≥ 65 years—United States, 2014. *Morbidity and Mortality Weekly Report* Vol. 65, No. 37, 993–998, 2016.
- [2] Khan, S. S.; Hoey, J. Review of fall detection techniques: A data availability perspective. *Medical Engineering & Physics* Vol. 39, 12–22, 2017.
- [3] World Health Organization. *WHO Global Report on Falls Prevention in Older Age*. World Health Organization, 2008.
- [4] Igual, R.; Medrano, C.; Plaza, I. Challenges, issues, and trends in fall detection systems. *BioMedical Engineering OnLine* Vol. 12, No. 1, 1–24, 2013.
- [5] Zhang, Z.; Conly, C.; Athitsos, V. A survey on vision-based fall detection. In: *Proceedings of the 8th ACM International Conference on Pervasive Technologies Related to Assistive Environments*, Article No. 46, 2015.
- [6] Doukas, C. N.; Maglogiannis, I. Emergency fall incidents detection in assisted living environments utilizing motion, sound, and visual perceptual components. *IEEE Transactions on Information Technology in Biomedicine* Vol. 15, No. 2, 277–289, 2011.
- [7] Sree, J.; Umamakeswari, A.; Jenita, B. A survey on technical approaches in fall detection system. *National Journal of Physiology, Pharmacy and Pharmacology* Vol. 5, No. 4, 275–279, 2015.

- [8] Reyes-Ortiz, J. L.; Oneto, L.; Samà, A.; Parra, X.; Anguita, D. Transition-aware human activity recognition using smartphones. *Neurocomputing* Vol. 171, 754–767, 2016.
- [9] Hammerla, N.; Halloran, S.; Ploetz, T. Deep, convolutional, and recurrent models for human activity recognition using wearables. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence, 1533–1540, 2016.
- [10] Chelli, A.; Patzold, M. A machine learning approach for fall detection and daily living activity recognition. *IEEE Access* Vol. 7, 38670–38687, 2019.
- [11] Aslan, M.; Sengur, A.; Xiao, Y.; Wang, H. B.; Cevdet Ince, M.; Ma, X. Shape feature encoding via Fisher Vector for efficient fall detection in depth-videos. *Applied Soft Computing* Vol. 37, 1023–1028, 2015.
- [12] Chua, J. L.; Chang, Y. C.; Lim, W. K. A simple vision-based fall detection technique for indoor video surveillance. *Signal, Image and Video Processing* Vol. 9, No. 3, 623–633, 2015.
- [13] Iazzi, A.; Rziza, M.; Oulad Haj Thami, R.; Aboutajdine, D. A new method for fall detection of elderly based on human shape and motion variation. In: *Advances in Visual Computing. Lecture Notes in Computer Science, Vol. 10073*. Bebis, G. et al. Eds. Springer Cham, 156–167, 2016.
- [14] Kumar, D. P.; Yun, Y.; Gu, I. Y. Fall detection in RGB-D videos by combining shape and motion features. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 1337–1341 2016.
- [15] Rougier, C.; Meunier, J.; St-Arnaud, A.; Rousseau, J. Robust video surveillance for fall detection based on human shape deformation. *IEEE Transactions on Circuits and Systems for Video Technology* Vol. 21, No. 5, 611–622, 2011.
- [16] Harrou, F.; Zerrouki, N.; Sun, Y.; Houacine, A. An integrated vision-based approach for efficient human fall detection in a home environment. *IEEE Access* Vol. 7, 114966–114974, 2019.
- [17] Kepski, M.; Kwolek, B. Fall detection using ceiling-mounted 3D depth camera. In: Proceedings of the International Conference on Computer Vision Theory and Applications, Vol. 2, 640–647, 2014.
- [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 7th Symposium on Information and Communication Technology, 339–344, 2016.
- [19] Fan, K. B.; Wang, P.; Zhuang, S. Human fall detection using slow feature analysis. *Multimedia Tools and Applications* Vol. 78, No. 7, 9101–9128, 2019.
- [20] Ismail, M. M. B.; Bchir, O. Automatic fall detection using membership based histogram descriptors. *Journal of Computer Science and Technology* Vol. 32, No. 2, 356–367, 2017.
- [21] Fan, K. B.; Wang, P.; Hu, Y.; Dou, B. J. Fall detection via human posture representation and support vector machine. *International Journal of Distributed Sensor Networks* Vol. 13, No. 5, 155014771770741, 2017.
- [22] Núñez-Marcos, A.; Azkune, G.; Arganda-Carreras, I. Vision-based fall detection with convolutional neural networks. *Wireless Communications and Mobile Computing* Vol. 2017, Article ID 9474806, 2017.
- [23] Akagunduz, E.; Aslan, M.; Sengu, A.; Wang, H. B.; Ince, M. C. Silhouette orientation volumes for efficient fall detection in depth videos. *IEEE Journal of Biomedical and Health Informatics* Vol. 21, No. 3, 756–763, 2017.
- [24] Abdulmunem, A.; Lai, Y. K.; Sun, X. F. Saliency guided local and global descriptors for effective action recognition. *Computational Visual Media* Vol. 2, No. 1, 97–106, 2016.
- [25] Vishwakarma, V.; Mandal, C.; Sural, S. Automatic detection of human fall in video. In: *Pattern Recognition and Machine Intelligence Lecture Notes in Computer Science, Vol. 4815*. Ghosh, A.; De, R. K.; Pal, S. K. Eds. Springer Berlin Heidelberg, 616–623, 2007.
- [26] Akagündüz, E. Shape recognition using orientational and morphological scale-spaces of curvatures. *IET Computer Vision* Vol. 9, No. 5, 750–757, 2015.
- [27] Yun, Y. X.; Gu, I. Y. 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* Vol. 148, 111–122, 2016.
- [28] Wiskott, L.; Sejnowski, T. J. Slow feature analysis: Unsupervised learning of invariances. *Neural Computation* Vol. 14, No. 4, 715–770, 2002.
- [29] Krizhevsky, A.; Sutskever, I.; Hinton, G. E. ImageNet classification with deep convolutional neural networks. *Communications of the ACM* Vol. 6, No. 6, 84–90, 2012.
- [30] Stauffer, C.; Grimson, W. E. L. Adaptive background mixture models for real-time tracking. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, 246–252, 1999.
- [31] Kim, K.; Chalidabhongse, T. H.; Harwood, D.; Davis, L. Background modeling and subtraction by codebook construction. In: Proceedings of the International Conference on Image Processing, Vol. 5, 3061–3064, 2004.

- [32] McFarlane, N. J. B.; Schofield, C. P. Segmentation and tracking of piglets in images. *Machine Vision and Applications* Vol. 8, No. 3, 187–193, 1995.
- [33] Yu, M.; Yu, Y. Z.; Rhuma, A.; Naqvi, S. M. R.; Wang, L.; Chambers, J. A. An online one class support vector machine-based person-specific fall detection system for monitoring an elderly individual in a room environment. *IEEE Journal of Biomedical and Health Informatics* Vol. 17, No. 6, 1002–1014, 2013.
- [34] Minaee, S.; Wang, Y. An ADMM approach to masked signal decomposition using subspace representation. *IEEE Transactions on Image Processing* Vol. 28, No. 7, 3192–3204, 2019.
- [35] Bakkay, M. C.; Rashwan, H. A.; Salmane, H.; Khoudour, L.; Puig, D.; Ruichek, Y. BSCGAN: Deep background subtraction with conditional generative adversarial networks. In: Proceedings of the 25th IEEE International Conference on Image Processing, 4018–4022, 2018.
- [36] Gomes, V.; Barcellos, P.; Scharcanski, J. Stochastic shadow detection using a hypergraph partitioning approach. *Pattern Recognition* Vol. 63, 30–44, 2017.
- [37] Gonzalez, R. C.; Woods, R. E.; Masters, B. R. Digital image processing third edition. *Journal of Biomedical Optics* Vol. 14, No. 2, 029901, 2008.
- [38] Yang, L.; Ren, Y. Y.; Zhang, W. Q. 3D depth image analysis for indoor fall detection of elderly people. *Digital Communications and Networks* Vol. 2, No. 1, 24–34, 2016.
- [39] Yu, M.; Rhuma, A.; Naqvi, S. M.; Wang, L.; Chambers, J. A posture recognition-based fall detection system for monitoring an elderly person in a smart home environment. *IEEE Transactions on Information Technology in Biomedicine* Vol. 16, No. 6, 1274–1286, 2012.
- [40] Minaee, S.; Boykov, Y.; Porikli, F.; Plaza, A.; Kehtarnavaz, N.; Terzopoulos, D. Image segmentation using deep learning: A survey. *arXiv preprint arXiv:2001.05566* 2020.
- [41] Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 39, No. 12, 2481–2495, 2017.
- [42] Farnebäck, G. Two-frame motion estimation based on polynomial expansion. In: Proceedings of the 13th Scandinavian Conference on Image Analysis, 363–370, 2003.
- [43] Bouguet, J. Y. Pyramidal implementation of the Lucas Kanade feature tracker description of the algorithm. Intel Corporation Microprocessor Research Labs.
- [44] Chang, C.-C.; Lin, C.-J. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* Vol. 2, No. 3, Article No. 27, 2011.
- [45] Bae, T. M.; Jin, S. H.; Ro, Y. M. Video segmentation using hidden Markov model with multimodal features. In: *Image and Video Retrieval. Lecture Notes in Computer Science*, Vol. 3115. Enser, P.; Kompatsiaris, Y.; O'Connor, N. E.; Smeaton, A. F.; Smeulders, A. W. M. Eds. Springer Berlin Heidelberg, 401–409, 2014.
- [46] Zhai, Y.; Shah, M. Video scene segmentation using Markov chain Monte Carlo. *IEEE Transactions on Multimedia* Vol. 8, No. 4, 686–697, 2006.
- [47] Charfi, I.; Miteran, J.; Dubois, J.; Atri, M.; Tourki, R. Definition and performance evaluation of a robust SVM based fall detection solution. In: Proceedings of the 8th International Conference on Signal Image Technology and Internet Based Systems, 218–224, 2012.
- [48] Auvinet, E.; Rougier, C.; Meunier, J.; St-Arnaud, A.; Rousseau, J. Multiple cameras fall dataset. Technical Report No. 1350. DIRO-Université de Montréal, 2010.
- [49] Khan, S.; Vizzari, G.; Bandini, S.; Basalamah, S. Detecting dominant motion flows and people counting in high density crowds. *Journal of WSCG* Vol. 22, Nos. 1–2, 21–30, 2014.

Abderrazak Iazzi received his bachelor of mathematics and computer science (MIT) degree in 2011 and his master degree in computer science and telecommunications in 2013 from the Faculty of Sciences, Rabat, Morocco. He is currently pursuing his Ph.D. degree in the Laboratory of Computer Science and Telecommunications (LRIT) at Mohammed V University, Rabat, Morocco, where he is studying fall detection for the elderly by exploiting video information.

Mohammed Rziza received his national doctorate in engineering sciences, with image processing speciality, from the Faculty of Science of the Mohammed V-Agdal University, Rabat, Morocco, in 2002. He joined the Faculty of Science, Rabat, Morocco, in 2003 first as an assistant professor, and then as an associate professor in 2010. Since 1997, he has been a member of the LRIT Laboratory and full professor since 2015 at Mohammed V University, Faculty of Sciences. His research interests include image processing, pattern recognition, and computer vision.

Rachid Oulad Haj Thami received his Ph.D. degree in computer science from the Faculty of Sciences Ben MSik Sidi Otthman, Casablanca, Morocco, in 2002. He is currently a full professor of computer engineering with the Higher National School of Computer Science and Systems Analysis (ENSIAS), Rabat

IT Center, Mohammed V University, Rabat, Morocco. His research interests include multimedia and information retrieval, image and video analysis, intelligent video surveillance, and health applications.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate

credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made.

The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Other papers from this open access journal are available free of charge from <http://www.springer.com/journal/41095>. To submit a manuscript, please go to <https://www.editorialmanager.com/cvmj>.