

Article

Fall Detection of Elderly People Using the Manifold of Positive Semidefinite Matrices

Abdessamad Youssfi Alaoui ^{1,*}, Youness Tabii ¹, Rachid Oulad Haj Thami ¹, Mohamed Daoudi ^{2,3}, Stefano Berretti ⁴ and Pietro Pala ⁴

¹ ADMIR Laboratory, Rabat IT Center, IRDATeam, ENSIAS, Mohammed V University in Rabat, Rabat 10000, Morocco; youness.tabii@ensias.um5.ac.ma (Y.T); rachid.ouladhajthami@gmail.com or rachid.ouladhajthami@ensias.um5.ac.ma (R.O.H.T)

² MT Lille Douai, Institut Mines-Télécom, Centre for Digital Systems, F-59000 Lille, France; mohamed.daoudi@imt-lille-douai.fr

³ CNRS, Centrale Lille, Institut Mines-Télécom, UMR 9189 CRISTAL, University Lille, F-59000 Lille, France

⁴ Department of Information Engineering, University of Florence, 50121 Florence, Italy; stefano.berretti@unifi.it (S.B.); pietro.pala@unifi.it (P.P.)

* Correspondence: abdessamad.youssfi@unifi.it

Abstract: Falls are one of the most critical health care risks for elderly people, being, in some adverse circumstances, an indirect cause of death. Furthermore, demographic forecasts for the future show a growing elderly population worldwide. In this context, models for automatic fall detection and prediction are of paramount relevance, especially AI applications that use ambient, sensors or computer vision. In this paper, we present an approach for fall detection using computer vision techniques. Video sequences of a person in a closed environment are used as inputs to our algorithm. In our approach, we first apply the V2V-PoseNet model to detect 2D body skeleton in every frame. Specifically, our approach involves four steps: (1) the body skeleton is detected by V2V-PoseNet in each frame; (2) joints of skeleton are first mapped into the Riemannian manifold of positive semidefinite matrices of fixed-rank 2 to build time-parameterized trajectories; (3) a temporal warping is performed on the trajectories, providing a (dis-)similarity measure between them; (4) finally, a pairwise proximity function SVM is used to classify them into fall or non-fall, incorporating the (dis-)similarity measure into the kernel function. We evaluated our approach on two publicly available datasets URFD and Charfi. The results of the proposed approach are competitive with respect to state-of-the-art methods, while only involving 2D body skeletons.

Keywords: fall detection; healthcare; positive semidefinite matrices; Riemann manifold; Dynamic Time Warping; Gram matrix



Citation: Youssfi Alaoui, A.; Tabii, Y.; Oulad Haj Thami, R.; Daoudi, M.; Berretti, S.; Pala, P. Fall Detection of Elderly People Using the Manifold of Positive Semidefinite Matrices. *J. Imaging* **2021**, *7*, 109. <https://doi.org/10.3390/jimaging7070109>

Academic Editors: Gonzalo Pajares Martinsanz and Raimondo Schettini

Received: 20 April 2021

Accepted: 23 June 2021

Published: 6 July 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In 2019, the United Nations (UN) published statistics about the world population [1]. According to this report, in the next years, the percentage of elderly people will grow considerably in Sub-Saharan Africa, Northern Africa, Western Asia, Latina America, Caribbean, Australia, North America, etc. In the same document, the estimated change in the percentage of elderly people between 2019 and 2050 is also reported. For example, the number of persons over 65 years in Morocco is expected to increase from 7.3% of population in 2019 to 11.2% of population in 2030.

On the other hand, the World Health Organization (WHO) published a report about the problems caused by falls [2]. This article reports an impressive statistic, according to which it is expected that most unintentional injuries in elderly people will be caused by falls. Another statistic shows that more than (646,000) persons die every year as a consequence of falling, and elderly people contribute the highest percentage of these deaths. They expect more than 37.3 million falls, the majority of them causing serious problems to healthcare

services. Furthermore, the Center for Disease Control Prevention [3] showed statistics about adult and senior falls. It results that about 20% of falls caused serious consequences, e.g., fractures, head injuries or hip fractures (in more than 95% of falls). Overall, more than 3 million seniors enter emergency departments every year due to falls.

In the last few years, a lot of solutions have been developed to decrease the danger caused by falls. For example, there are a lot of works aiming to monitor persons by using cameras, sensors and ambient/fusion [4–7]. These methods analyze the motion of persons and aim to distinguish between falls and daily activities.

Several approaches use wearable sensors, such as accelerometers and gyroscopes to detect posture and inactivity of the person [8–15] and extract different features from the data: angles, directions, acceleration, etc. The classification step is typically performed by using thresholds or machine learning algorithms. Other works based their models on utilizing room information, such as sound and floor vibration [16,17]. Generally, it is very easy to setup a fall detection system using a wearable device-based approach. However, these systems are expensive, not robust and consume batteries. Moreover, the accuracy and intrusion of these methods depend on the specific scenarios. Last, it is impossible for the care services to visualize and verify the data to better understand and improve the obtained accuracy.

Computer-vision approaches monitor an imaged subject by using cameras [18–31]. They analyze the change of body shape by computing different features, such as the ratio between height and width of the box surrounding the person, the histogram projection of the silhouette, the coordinates of an ellipse surrounding the person and the key joints of the person's skeleton. Furthermore, computer-vision approaches are the most used to detect fall thanks to their robustness and ease of setting up a fall detection system. These methods are highly accurate and do not depend on scenarios. In Section 2, we present more details about previous fall detection works appeared in the literature.

In this paper, we present an algorithm to detect fall using a computer-vision approach that does not rely on wearable sensors or handheld devices. Firstly, we apply the V2V-PoseNet [32] model to detect the skeleton of the person in a 2D image. In the second step, we measure the similarity between sequences of skeletons and use the matrix of similarity scores between all sequences as an input to a classifier. In doing so, we rely on the method in [33] and employ the Riemannian manifold of positive semidefinite matrices to compute a trajectory from the skeletons of the person. Then, we employ the Dynamic Time Warping algorithm to align the trajectories and compute the similarity scores between sequences. Finally, we use a Support Vector Machine (SVM) to classify between fall and non-fall events using the similarity scores.

The rest of the paper is organized as follow. In Section 2, we summarize works in the literature that detect fall by using ambient/fusion, wearable sensors and computer vision. In Section 3, we show the different operations that constitute our approach. In Section 4, we present the results of applying our solution by using the Chari and the URFD datasets and compare with state-of-the-art methods. Finally, in Section 5, we conclude our paper and present some perspectives for future work.

2. Related Work

In the last few years, many approaches have been proposed to detect fall of elderly people [4–7]. These works can be grouped in three main categories [4]: (i) methods that use wearable sensors to monitor the person and detect abnormal activities during time; (ii) solutions that use ambient/fusion to collect room information such as floor vibration and sound, with recent works that also utilize other technologies such as smartphones, Wi-Fi, etc.; and (iii) methods that use a camera to detect the change of body shape during time. In the following, we focus more on the methods in the third category since they are closer to our proposed approach.

Wearable sensors: Wearable device-based approaches use triaxial gyroscopes [12,13], accelerometers [8–11,15,34,35] or both types of sensors [36] to monitor the person and

detect posture changes and inactivity. In these solutions, data acquired by the sensors are used to compute different features, such as angles [9,12], differences and derivatives of the sum X , Y and directions [8,9], maximum acceleration and fluctuation frequency [12], decreasing of heat rates [10], variation of different parts of the body [11], the acceleration of the body parts [13], mutual information and removing highly correlated features using Pearson correlation coefficient and Boruta algorithm [35], etc. In addition, they distinguish fall and non-fall events by using thresholds [8–10], machine learning [11–14,35] and deep learning algorithms [15,34].

Ambient/fusion: Many works used the sound captured in the environment as a clue for detecting the fall of a person [16,17]. This is obtained by detecting the sound of the person during fall and normal activities, in order to compute the Mel-frequency spectral coefficient. In the last step, fall and non-fall events are classified by using machine learning techniques.

Computer vision-based: Many methods have been developed to monitor a person using cameras. Sequences of frames are used to calculate different features such as histogram projection of the person's silhouette [18,22,37]; aspect ratio and orientation of principal components [21]; motion vectors of the person [19,20], bounding box coordinates [22,24], feet-related features such as step length symmetry, normalized step count, speed and foot flat ratios [37]; body-related features such as amount of movement in the left and right side of the body, movement symmetry, shift in the centre of gravity and torso orientation [37]; etc. Other works employed Riemannian manifolds to analyze the shape of the person and detect fall [23,24]. In addition, solutions based on deep learning algorithms such as Convolution Neural Networks (CNNs) [19,20] and Long-Short Term Memory networks (LSTM) [38] have been also used.

Several methods exist that use the skeleton of the monitored person to compute features in every frame of a sequence. These methods can either detect the skeleton of a person in 2D images by using CNN models, such as OpenPose, PoseNet, ALphaNet, etc., or they can detect the skeleton in images captured by a Kinect sensor. Relying of the detected skeleton, several methods estimate the human pose by extracting features from the skeleton and classifying them. For example, Chen et al. [25] developed an algorithm to recognize accidental falls by using the skeleton information. They first detected the skeleton of the person by applying the OpenPose algorithm. Then, they computed the speed of descent of the hip joint's center, the angle between the floor and the center-line of the human body and the ratio between the width and height of the rectangle surrounding the human body. They take on consideration the standing up of the person after fall in their algorithm. Their model achieved a success rate in fall recognition of 97%. Alaoui et al. [39] developed an algorithm to detect falls by using the variation of a person's skeleton into the video. Firstly, they detected the joints of the person into the video by using OpenPose. Then, they computed the angle and the distance between the same joint into two sequential frames. Finally, they trained their model using SVM, KNN, Decision Tree and Random Forest to classify fall and non-fall sequences. The SVM classifier resulted the most effective in their work. Loureiro and Correia [40] employed the VGG-19 architecture to classify pathological gaits or to extract features from the skeleton energy image. After that, they fed these features into a linear Discriminant Analysis Model and Support Vector Machine to classify normal, diplegic and hemiplegic gaits simulated by healthy people.

There are many methods that use images captured by Kinect sensors, in order to generate joint positions of the human's body. For example, Yao et al. [26] developed an algorithm that includes three steps. Firstly, they captured motion information of joints in 3D coordinates using the R, G and B channels of a pixel. They focused on a reduced set of 25 joints of the human skeleton. Then, every frame is encoded independently as a slice of motion image, in order to overcome the problem of losing information caused by trajectory overlap. In the last step, the Limit Of Stability Test (LOST) is used to detect fall from the start to the end key frame. They reported an accuracy of 97.35% on the TST v2 dataset, with effective performed reported also on the UT-A3D dataset. Kawatsu et al. [27]

proposed an approach to detect fall. They developed two algorithms that use the skeleton generated from the Kinect SDK. The first algorithm aims to determinate the maximum distance between the floor and the position of all the joints. They detected fall by comparing this distance with a threshold. The second algorithm computes the average velocity of all the joints. In addition, in this case, to detect falls, the average velocity is compared with a threshold.

Alazrai et al. [28] developed a fall detection algorithm based on a representation layer and two classification layers. They used a Kinect sensor to collect RGBD images and derive 3D joint positions. They computed the Motion Pose Geometric Descriptor (MPGD) for every input frame in order to describe motion and pose of human body parts. After that, they employed an SVM to classify every frame in the first classification layer. The second classification layer employed the Dynamic Time Warping algorithm to classify fall and non-fall sequences generated from the SVM. They tested the model by using the 66-dataset that contains 14,400 frames and 180 activity sequences. Using five-fold cross validation, they achieved 98.01% precision, 97.13% recall and 97.57% F1-measure. Pathak et al. [29] proposed a fall detection method. They first detected and tracked key joints from a Kinect sensor and extracted two parameters using key joints. Then, to detect falls, they compared these parameters with a threshold. They also integrated in their system an alert message, which is sent to a predefined number when a fall event is detected. They tested the model on a real dataset of 50 persons, obtaining 94.65% accuracy in indoor environment. Abobakret et al. [31] presented an algorithm to detect fall using a skeleton posture and activity recognition. They analyzed local variations in depth pixels to recognize the posture using frame acquired from a Kinect-like sensor. They employed a random decision forest to distinguish standing, sitting and fall postures and detected fall events by employing an SVM classifier. They reported 99% sensitivity on a synthetic live dataset, 99% sensitivity on a synthetic dataset and 96% sensitivity on popular live dataset without using accelerometer support. Seredin et al. [30] developed an algorithm to detect falls by using skeleton feature encoding and SVM. They computed a cumulative sum to combine the decision on a sequence of frames. The model achieved 95.8% accuracy in the cross validation procedure, using a Leave-One-Person-Out protocol.

Discussion: As summarized above, different methods exist for fall detection. These algorithms have been evaluated using their sensitivity and specificity, which resulted highly effective in many cases. For example, algorithms that employ depth sensors are very accurate. In addition, systems reported a high accuracy when they employed multi-dimensional combination between physiological and kinematic features [26–31]. However, existing solutions show several limitations. For example, systems that use Wearable [8–15,34–36] and ambient [16,17] features have some disadvantages, which are related to the inconceivability of visually checking object information. In addition, systems that use computer vision techniques [18–25] are flexible. The majority of these algorithms are not specific, do not depend on different scenarios, are simple to setup and are very accurate [4].

3. Proposed Method

We present a method to classify between fall and non-fall events, which is based on computing the similarity between video sequences. Figure 1 provides an overview of the proposed approach. As a preliminary step, we employ the V2V-PoseNet [32] model to extract a skeleton from each frame of a sequence. After that, we represent our data (i.e., the sequence of skeletons) using Gram matrices and thus on the Riemannian manifold of positive semidefinite matrices. To this end, we compute a Gram matrix from the skeleton in each frame of a sequence. Gram matrices are symmetric matrices that lay on the Riemannian manifold of positive semidefinite matrices, so that a sequence is transformed to a trajectory of points on the manifold, i.e., a point is derived on the manifold for each frame. A Riemannian metric is then defined on the manifold to compare two Gram matrices. Finally, the Dynamic Time Warping (DTW) algorithm is used to extend the Riemannian metric from the frame level to the sequence level and compute a similarity score between two

sequences. This aims to be invariant with respect to differences in the speed of execution of the action captured in a sequence. This score is the input to a linear SVM classifier that we use to distinguish between fall and non-fall events.

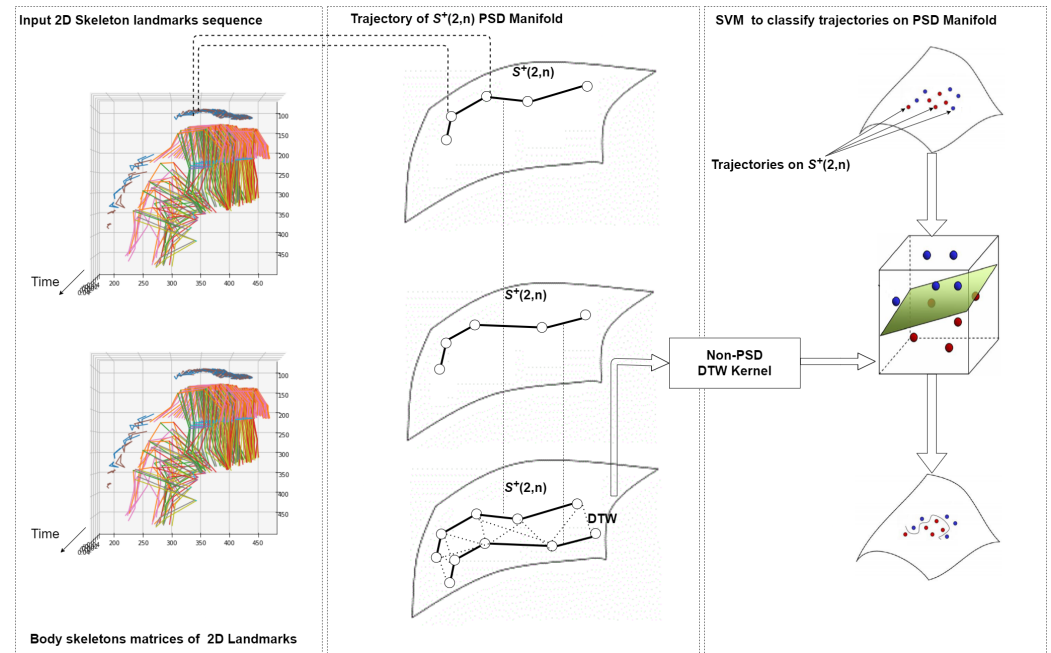


Figure 1. Overview of the proposed approach. After detecting the skeleton of a person in every frame of a sequence, a Gram matrix is computed for each frame. In this way, a sequence is represented by a trajectory of points on the manifold of positive semidefinite matrices $S^+(2, n)$. The Dynamic Time Warping (DTW) algorithm is employed to align trajectories on the manifold. Finally, a kernel generated from DTW and linear SVM are employed to classify fall and non-fall sequences.

3.1. Skeleton of a Person

The skeleton of a person is detected by using the V2V-PoseNet [32] model for each frame of a sequence. There are four steps in the construction of the V2V-PoseNet model. First, a volumetric convolution is computed by utilizing a volumetric basic block, then a volumetric batch normalization [41] plus an activation operation are applied to remove negative values. In the second step, a volumetric residual block is employed to extend from a 2D convolution block. The residual blocks exploit the result of the previous convolution blocks in order to extract more features. The third step applies a down-sampling operation using max-pooling, thus reducing the image dimension and the processing time. In the last step, a volumetric decoding block is used to decode the results of the previous steps (i.e., decode features found in the previous steps and visualize them into the input images). This step also contains a volumetric normalization block and an activation operation (ReLU) to remove the negative values as well as to normalize the values produced after the decoder block. In summary, the V2V-PoseNet [32] model applies convolutional blocks and computes features, producing the pose confidence and joint key points of the person in a given image as initial result. Then, it visualizes these features into the input image as key points. We also tested other algorithms to detect the skeleton of the person such as OpenPose and AlphaNet. We found that V2V-PoseNet is the most accurate to detect skeletons from videos. Figure 2 illustrates the results of detecting skeletons by using V2V-PoseNet and the 3D projection of a sequence of skeletons detected from a video sequence.

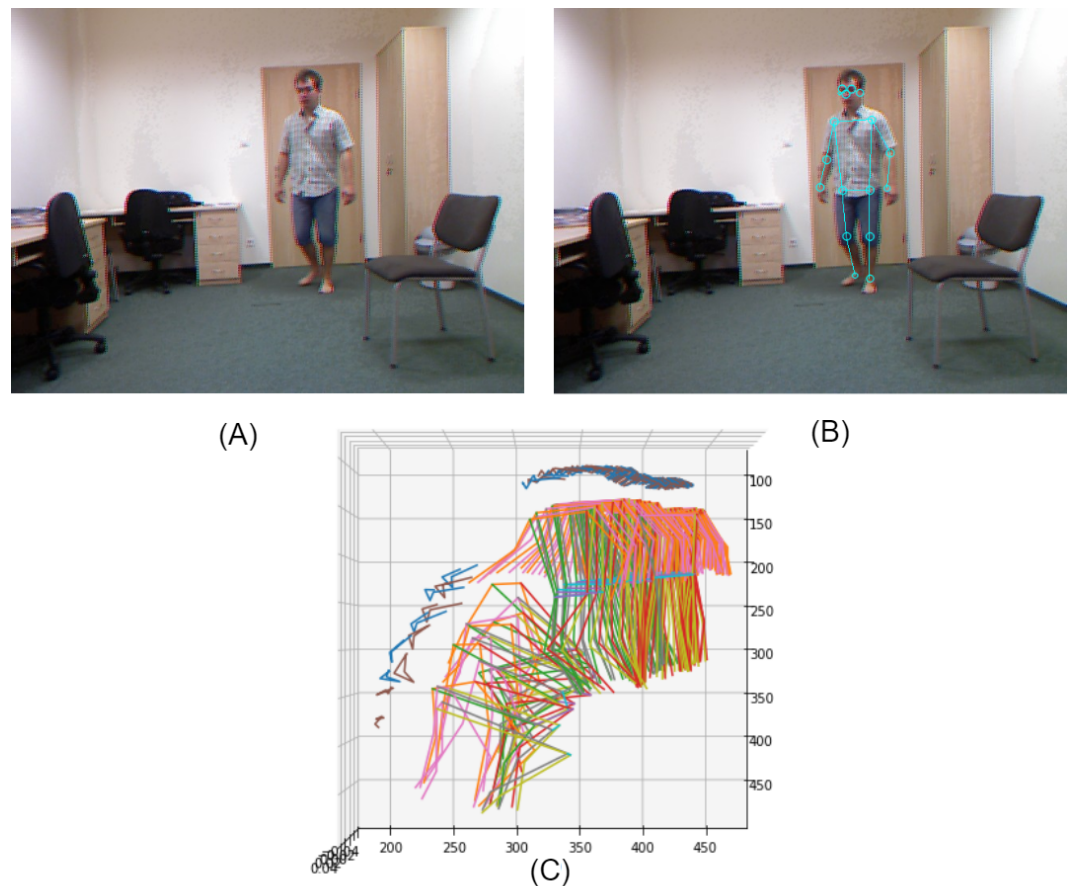


Figure 2. Results of applying the V2V-PoseNet model to detect the skeleton of a person: (A) the input frame to V2V-PoseNet; (B) projection into the input image of the skeleton detected by V2V-PoseNet; and (C) the 3D projection of a sequence of skeletons corresponding to a video sequence.

In our work, we aim at detecting falls by analyzing the change of a person’s body during a sequence. To this end, we extract the skeleton of the person and represent the shape by a set of points. In this way, the shape is given by a time series of the 2D coordinates of the points. This time series contains the coordinates of all the skeletons tracked during an event. A fall event is detected from a sequence containing m frames, where every frame is represented by a vector with the skeleton coordinates, i.e., the vector V_i contains $\{(x_1, y_1), \dots, (x_n, y_n)\}$, where n is the number of joints. Every video is characterized by a set of vectors, where every vector represents the coordinates of skeleton’s points (i.e., a video corresponds to V_1, \dots, V_m). More specifically, every $V_{1 \leq i \leq m}$ is a $n \times 2$ matrix.

We represent a sequence of skeletons by a sequence of Gram matrices, where each Gram matrix is computed as:

$$G_i = V_i V_i^T . \tag{1}$$

The resulting matrix is an $n \times n$ positive semidefinite matrix, with a rank smaller than or equal to 2. Such $n \times n$ positive semidefinite matrices of rank 2 have been studied in several works [42–48].

These Gram matrices belong to $S^+(2, n)$, the manifold of $n \times n$ positive semidefinite matrices of rank 2 for which a valid Riemannian metric can be defined as follows:

$$d(G_1, G_2) = \left[\text{tr}(G_1) + \text{tr}(G_2) - 2 \text{tr} \left(\left(G_1^{\frac{1}{2}} G_2 G_1^{\frac{1}{2}} \right)^{\frac{1}{2}} \right) \right]^{\frac{1}{2}} , \tag{2}$$

being G_1 and G_2 two generic Gram matrices in $S^+(2, n)$. We can also use Singular Value Decomposition (SVD) to compute the previous distance by employing:

$$d(G_1, G_2) = \min_{Q \in Q^d} \|V_1 Q - V_2\|_F . \tag{3}$$

The optimal value $Q^* = AU$ is computed by using SVD, where $V_1^T V_2 = A \Sigma U^T$.

Our final goal is classifying fall and non-fall sequences by using the similarity scores. This requires for a method computing a similarity score between sequences, as described in the following.

3.2. Sequence Similarity Using the DTW Algorithm

Dynamic Time Warping (DTW) is an algorithm that aims at measuring the similarity between two temporal sequences. As such, it is largely employed in time series analysis. One important characteristic of DTW is its capability of computing the similarity between two sequences that vary in speed, so with different acceleration or deceleration. For example, two sequences can correspond to two persons walking with different velocities.

In fall detection, the major difference between a fall event and a non-fall event is the acceleration and deceleration of a person. The acceleration of a falling person is bigger than the acceleration of a person who is not falling. For this reason, we employ DTW to measure the similarity between sequences [49,50].

As discussed above, we represent a sequence of skeletons as a sequence of Gram matrices. For example, let $V^1 = \{V_0^1, \dots, V_{\tau_1}^1\}$ and $V^2 = \{V_0^2, \dots, V_{\tau_2}^2\}$ be two sequences of skeleton matrices. Computing the Gram matrices, we represent $V^1 = \{V_0^1, \dots, V_{\tau_1}^1\}$ by $G^1 = \{G_0^1, \dots, G_{\tau_1}^1\}$, where $G_i^1 = V_i^1 V_i^{1T}$, $0 \leq i \leq \tau_1$, and $V^2 = \{V_0^2, \dots, V_{\tau_2}^2\}$ by $G^2 = \{G_0^2, \dots, G_{\tau_2}^2\}$, where $G_j^2 = V_j^2 V_j^{2T}$ and $0 \leq j \leq \tau_2$. Then, we compute the distance between any two pairs of Gram matrices in the two sequences. The result is a matrix of size $\tau_1 \times \tau_2$, where $D(i, j)$ is the distance between G_i^1 and G_j^2 (i.e., V_i^1 and V_j^2 , respectively), $0 \leq i \leq \tau_1$ and $0 \leq j \leq \tau_2$. $D(i, j)$ is computed as:

$$D(i, j) = d(G_i^1, G_j^2) . \tag{4}$$

where $d(\cdot, \cdot)$ is the Riemannian metric defined in (2). The matrix D is used as input to the DTW algorithm that computes the distance $D_{DTW}(G^1, G^2)$ between the two sequences of Gram matrices.

The result of this computation is used to evaluate the Gaussian kernel required to train the SVM classifier. For two generic sequences i and j , this is defined as:

$$k(i, j) = \frac{1}{2} \times \exp - \frac{D_{DTW}(i, j)}{2\sigma^2} . \tag{5}$$

The DTW algorithm that we use here is based on the work of Gudmundsson et al. [51]. The Algorithm 1 summarizes the computation of the Gaussian Kernel using the Riemannian distance between two sequences.

Algorithm 1: Gaussian Kernel.

input : Two sequences of skeletons matrices $V^1 = \{V_0^1, \dots, V_{\tau_1}^1\}$ and $V^2 = \{V_0^2, \dots, V_{\tau_2}^2\}$

output: Matrix k contains the results values of applying Gaussian Kernel between two sequences V^1 and V^2

$k \leftarrow \text{Zeros}(\tau_1, \tau_2)$;

for $i \leftarrow 1$ **to** τ_1 **do**

for $j \leftarrow 1$ **to** τ_2 **do**

$k(i, j) \leftarrow \frac{1}{2} \times \exp - \frac{D_{DTW}(i, j)}{2\sigma^2}$ (5)

end

end

In the DTW algorithm, a matrix M is computed, where the $M(i, j)$ element is the sum of $k(i, j)$ and the minimum of $M(i, j - 1) + M(i - 1, j - 1) + M(i - 1, j)$. The matrix M has size $(1 + \tau_1) \times (1 + \tau_2)$, where τ_1 is the size of the sequence V^1 , τ_2 is the size of the sequence V^2 , $1 \leq i \leq 1 + \tau_1$ and $1 \leq j \leq 1 + \tau_2$. In particular, the element $M(i, j)$ is computed as:

$$M(i, j) = k(i, j) + \min\{M(i, j - 1), M(i - 1, j - 1), M(i - 1, j)\} . \tag{6}$$

The similarity score between these two sequences is the last value of the matrix (i.e., $M(1 + \tau_1, 1 + \tau_2)$). The Algorithm 2 summarizes the DTW procedure.

Algorithm 2: Dynamic Time Warping.

input : Two sequences of skeletons matrices $V^1 = \{V_0^1, \dots, V_{\tau_1}^1\}$ and $V^2 = \{V_0^2, \dots, V_{\tau_2}^2\}$
output: $M(\tau_1 + 1, \tau_2 + 1)$, similarity value between V^1 and V^2
 $k \leftarrow \text{Algorithm 1}(V^1, V^2);$
 $M \leftarrow \text{Zeros}(\tau_1 + 1, \tau_2 + 1);$
for $i \leftarrow 1$ **to** $\tau_1 + 1$ **do**
 for $j \leftarrow 1$ **to** $\tau_2 + 1$ **do**
 $M(i, j) \leftarrow k(i, j) + \min(M(i, j - 1), M(i - 1, j - 1), M(i - 1, j))$ (6)
 end
end

3.3. Classification

Once the similarity scores have been computed, we use them as input to an SVM classifier. To this end, we represent every sequence by a vector, which contains the similarity scores between this sequence and all the other sequences. Let $V^i = \{v_0^i, \dots, v_{\tau_i}^i\}$ be a sequence of skeleton matrices, where τ_i is the number of skeleton matrices (respectively, $G^i = \{G_0^i, \dots, G_{\tau_i}^i\}$). The similarity vector is computed as $\{\phi(V^1, V^i), \dots, \phi(V^i, V^i), \dots, \phi(V^i, V^s)\}$, where s is the number of sequences, and $\phi(V^i, V^j)_{0 \leq i, j \leq s}$ is the similarity score (computed using the Gaussian kernel) between sequences V^i and V^j . Now, we can represent the set of sequences by a set of vectors containing the similarity scores. This results into a matrix X , where X_j is the j th line of matrix X and corresponds to the similarity scores between the j th and all others sequences (it also contains the similarity score between the j th sequence and itself).

The Algorithm 3 summarizes the computation of the similarity scores matrix.

Algorithm 3: Computation of the similarity scores matrix.

input : Set of sequences $S = \{V^1, \dots, V^s\}$, where s is the number of sequences.
output: Matrix X contains similarity scores
 $\phi \leftarrow \text{Algorithm 2}(V^1, V^2);$
 $X \leftarrow \text{Zeros}(\tau_1 + 1, \tau_2 + 1);$
for $i \leftarrow 1$ **to** s **do**
 for $j \leftarrow 1$ **to** s **do**
 $X(i, j) \leftarrow \phi(V^i, V^j)$
 end
end

4. Experimental Results

We evaluated the performance of our approach on the Charfi [52] and UR Fall Detection [53] datasets, and compared the results against state-of-the-art methods as reported below.

4.1. Charfi Dataset

This dataset was designed and acquired at the "Laboratoire Electronique, Informatique and Image" (Le2i) [52]. It includes 240 videos with resolution of 320×240 , each reporting fall and non-fall events as occurring in daily activities. The dataset includes several daily life activities, such as sitting, laying, sleeping and also falling events, while performed in different locations, including reading room, office, coffee and home. Various views of the camera are used to monitor the imaged person. The background of the video is fixed and simple, while the texture of images is difficult. Moreover, shadows are also present in this dataset. Every frame is labeled with the location, the fall/non-fall class and by the coordinates of the person given as bounding box. Figure 3 illustrates some frames from the Charfi dataset in different locations.

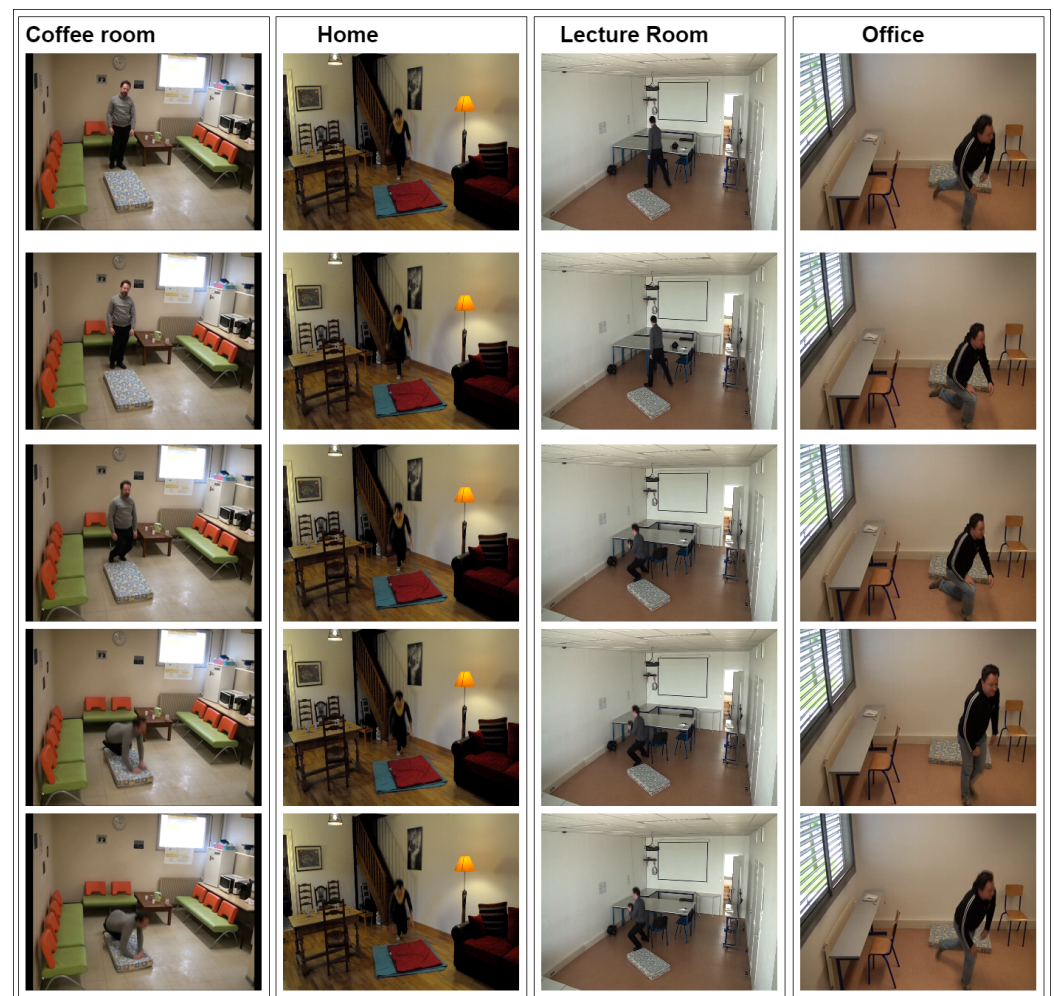


Figure 3. Charfi dataset: Examples frames taken from lecture room, home, coffee room and office locations.

Using the V2V-PoseNet, we first extract the skeleton joints of the person in the frames, as illustrated in Figure 4. Interestingly, the detection is robust to the presence of shadows in the image.

To evaluate the performance of our algorithm, we adopted a Leave-One-Out cross validation protocol as in [22,54]. According to this, the training set comprises all sequences except one, while the sequence left out is used as test. Using iteratively all the sequences of the Charfi dataset as test, we obtained the normalized confusion matrix reported in Table 1.

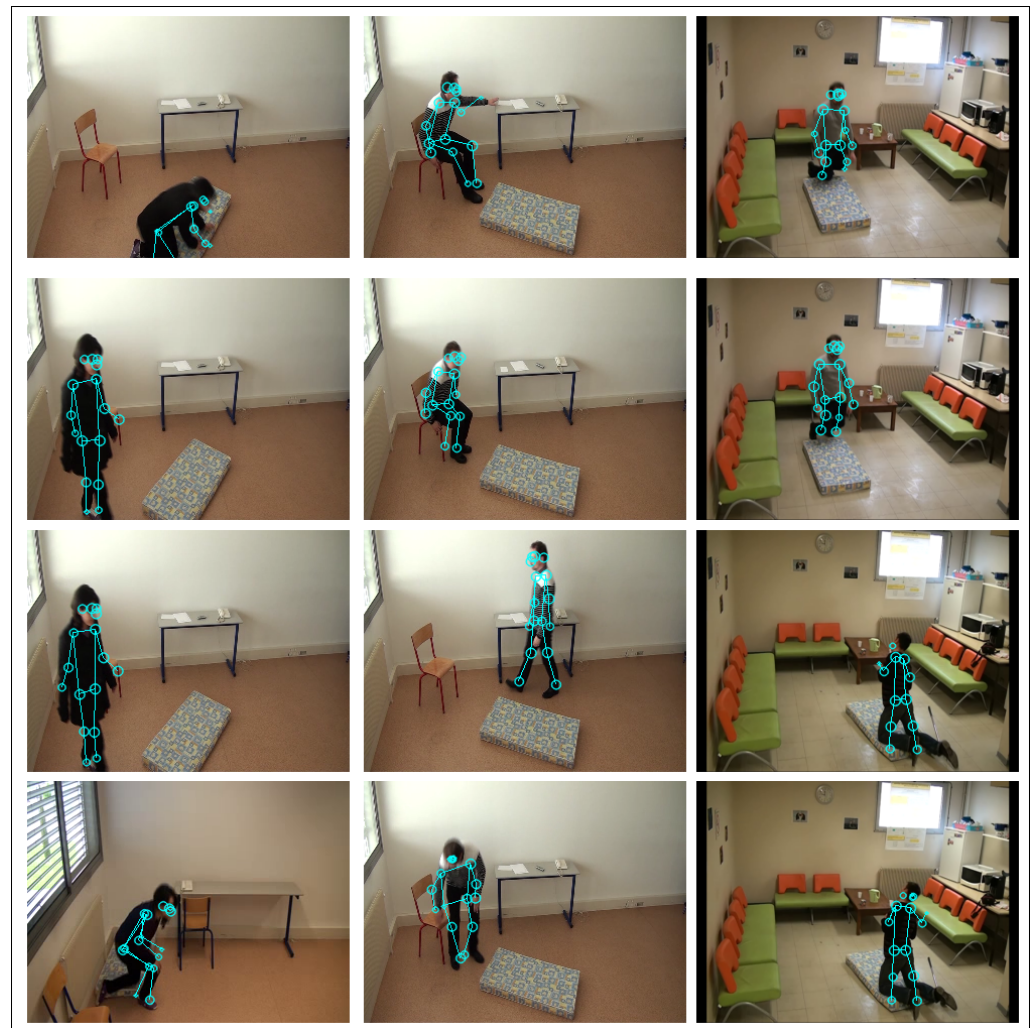


Figure 4. Charfi dataset: Skeleton detected by applying V2V-PoseNet to some frames.

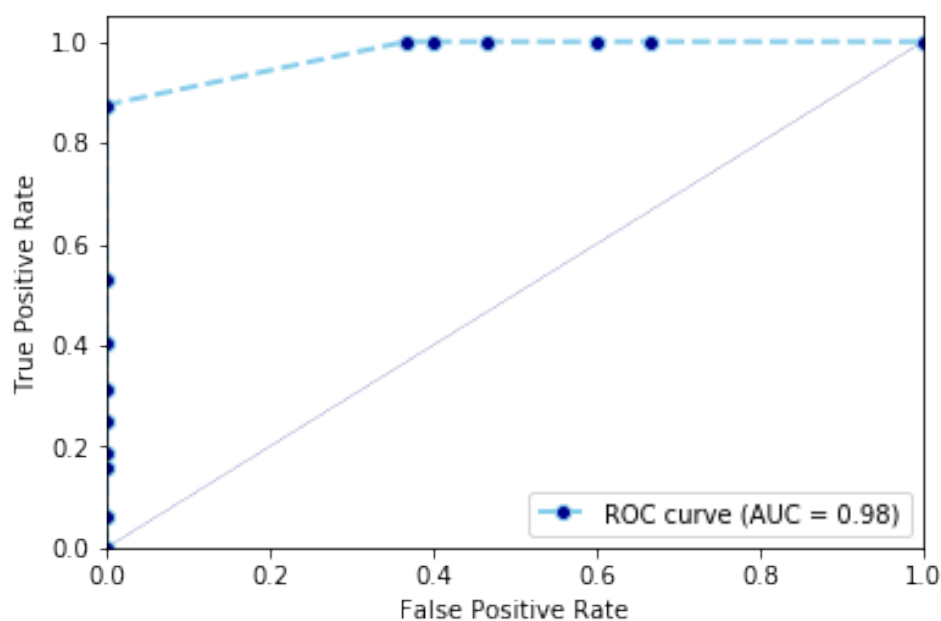
Table 1. Charfi dataset: The confusion matrix obtained by applying our approach.

	Predicted label	
	Fall	Non-Fall
Real label		
Fall	97	0
Non-Fall	19	131

Table 2 illustrates the accuracy, specificity and sensitivity of our algorithm and other works on the Charfi dataset. For example, applying our algorithm results into an accuracy of 93.67%, a specificity of 87% and a sensitivity of 100%. In particular, we observe in this dataset there are some videos corresponding to a normal activity of the person that are similar to a fall sequence. For these sequences, the speed of the person is similar to the speed of a person as occurring in fall sequences. This makes the detection challenging and results in a specificity (i.e., the capability of classifying a non-fall sequence that is similar to a fall) of 87%. In addition, our algorithm classifies between fall and non-fall sequences with an accuracy of 93.67%. This value represents the capacity of classifying between fall sequences and non-fall sequences classified as non-fall sequences. This value is represented also by the ROC curve in Figure 5, which represents the cumulative rate between true positives (i.e., fall sequences classified as fall sequences) and false positives (i.e., non-fall sequences classified as such).

Table 2. Charfi dataset: Sensitivity, specificity and accuracy of our work in comparison to state-of-the-art methods.

Methods	Sensitivity	Specificity	Accuracy
Georgios Goudelis et al. [54]	-	-	100–96.6
Charfi et al. [52]	73	97.7	-
M. Chamle et al. [55]	83.47	73.07	79.31
Arisa Poonsri et al. [21]	93	64.29	86.21
Alaoui et al. [56]	94.55	90.84	90.9
Alaoui et al. [39]	95	100	97.5
Ours	100	87	93.67

**Figure 5.** Charfi dataset: ROC curve representing the cumulative rate between true positive rate and false positive rate.

4.2. UR Fall Detection Dataset

The URFD dataset was designed and captured by the Interdisciplinary Center for Computational Modeling at the University of Rezesow [53]. The imaged person is monitored in a closed environment, in order to capture the maximum of person's activities. Daily activities were considered such as lying on the floor, crouching down, lying on the bed/couch, sitting down and picking up an object. Video sequences corresponding to fall events contain the person during fall, after fall (this part of the sequence is not used in the classification) and before fall. Figure 6 shows some example frames from the URFD dataset.

The results of extracting the skeleton joints of a person with the V2V-PoseNet model on some frame of the UR dataset are shown in Figure 7.

We evaluated the performance of using URFD by adopting a Leave-One-Out cross validation protocol. The concept of this validation protocol consists of using one sequence left out for testing, while all the remaining sequences are used for training. The normalized confusion matrix obtained by applying our approach on the URFD dataset is reported in Table 3.



Figure 6. URFD dataset: example frames.

Table 3. URFD dataset: The confusion matrix obtained by applying our algorithm.

		Predicted label	
		Fall	Non-Fall
Real label	Fall	63	0
	Non-Fall	2	27

Videos in the URFD dataset have a short duration and correspond to fall and non-fall sequences. For these videos, the speed of a person performing daily activities is not high, when compared to some videos in the Charfi dataset. This results in a higher specificity that reaches the value of 96.55%. This value is represented also using the Receiver Operating Characteristic (ROC) curve in Figure 8. The ROC curve is used to plot the probability of detecting fall by using similarity scores (the ratio between sequences, which are detected as fall sequences and all fall sequences) against probability of detecting non-fall sequences (the ratio between non-fall sequences, which are detected as fall sequences and all non-fall sequences) at various thresholds. The comparison between the results obtained by our

proposed method and approaches reported in the state-of-the-art on the URFD dataset is shown in Table 4. The speed of the person has an important impact on the classification. For this reason, our algorithm reaches 100% sensitivity for both datasets. The most critical challenge for our algorithm is the difference in acceleration (deceleration) values. For example, a person who sits down very quickly is probably classified as a fall sequence. Moreover, the skeleton of the person will not be detected in dark room. This can suggest a possible adaptation of our algorithm with very poor lighting, in order to overcome the problem of low illumination.



Figure 7. UR dataset: Skeleton detected with the V2V-PoseNet on some frames.

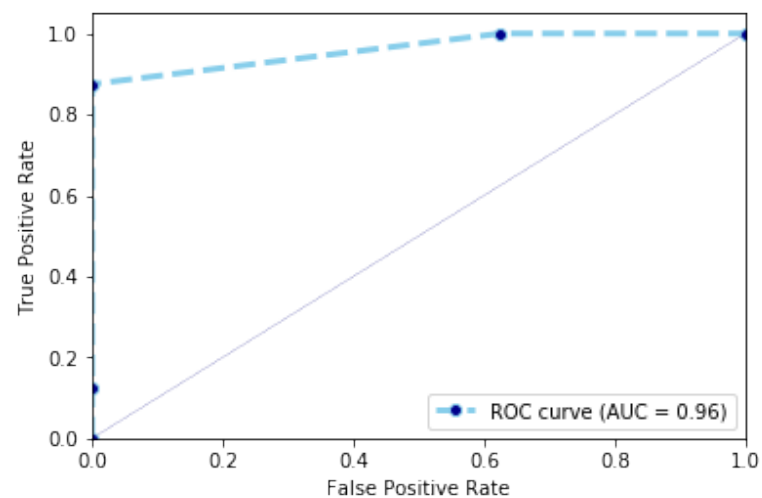


Figure 8. URFD dataset: ROC curve representing the cumulative rate between true positive rate and false positive rate.

Table 4. URFD dataset: Sensitivity, specificity and accuracy of our work in comparison to state-of-the-art methods.

Methods	Sensitivity	Specificity	Accuracy
Ali, Syed Farooq et al. [57]	99.03–99.13	99.03	-
Kepski et al. [58]	100	96.67	95.71
Bourke et al. [59]	100	90	-
Kepski et al. [53]	100	92.5	95
Alaoui et al. [39]	100	95	97.5
Yixiao Yun et al. [23]	96.77	89.74	-
Ours	100	93	96.55

4.3. Cross Data Evaluation

We also performed a cross-dataset evaluation. We trained our algorithm using sequences from the Charfi dataset. After that, we tested our model using sequences from the URFD dataset and vice versa. Table 5 illustrates the results of using the cross data evaluation protocol. Our algorithm has an accuracy of 87.39%, sensitivity of 100% and specificity of 62.5% using the Charfi dataset to train our algorithm and the URFD dataset to test it. In addition, our algorithm reports an accuracy of 85.34%, specificity of 62.85% and sensitivity of 95.06% by utilizing the URFD dataset to training and the Charfi dataset for testing. Our algorithm is able to detect fall sequences. For this reason, we found a high value for sensitivity. However, our method cannot detect non-fall sequences correctly. For this reason, we found a low value for specificity because of the differences between the length of non-fall sequences into the Charfi and URFD datasets. Charfi contains sequences with different conditions such as light and dark environment, different places, etc.; thus, we found that the accuracy using Charfi to train our model is greater than that using URFD.

Table 5. Cross data evaluation: Sensitivity, specificity and accuracy using the Charfi dataset to train our algorithm and the URFD dataset as testing and vice versa.

Training Dataset	Testing Dataset	Sensitivity	Specificity	Accuracy
Charfi	URFD	100	62.5	87.39
URFD	Charfi	95.06	62.85	85.34

4.4. Computation Time

In order to measure the overall processing time of our algorithm, we computed the time of every step included in our processing pipeline: (i) the time for extracting the skeleton from an individual frame; (ii) the time for applying the DTW algorithm between two sequences; and (iii) the time for the classification step (i.e., time to classify fall and non-fall sequences by a linear SVM). These times were computed on a laptop equipped with a i5-7200U (7th gen) processor, 8 GB of RAM and 2 GB NVIDIA GeForce 940MX.

Table 6 presents the results of computing the processing time for each step. We used two images to evaluate the time of the V2V-PoseNet step. For the first image taken from the Charfi dataset, we measured a time of 0.277 s to extract the skeleton. For the same step, but performed on an image from the URFD dataset, the skeleton detection operation required 0.32 s. This difference can be explained by the different resolution of the images in the Charfi and URFD datasets. It is relevant to note here that V2V-PoseNet is capable of detecting the skeleton without problem due to shadow.

The time for the DTW step was computed using two sequences from the Charfi dataset, the first one with 12 frames and the second containing 32 frames. The resulting time was 0.063 s. For the URFD dataset, we computed the DTW step for two sequences, with 12 and 30 frames, respectively, with a processing time of 0.061 s. The last step consists of classifying the similarity score vectors using sequences from the URFD dataset. The time required by this step was 0.053 s. For the Charfi dataset, the time taken for the classifying

fall and non-fall sequences was 0.65 s. It can be noticed that the processing time is low, yet the approach is theoretically solid and robust.

Table 6. Computation time (in seconds) for each step of our algorithm. Computation times were computed separately for the Charfi and URFD datasets.

Dataset	V2V-PoseNet	DTW	Classification (Linear SVM)
URFD	0.32	0.061	0.053
Charfi	0.277	0.063	0.65

4.5. Discussion

The proposed method aims to detect falls using a computer-vision approach. The major advantage of using a camera to monitor a person is overcoming the problem of background noise in the environment that is observed when using wearable sensors [8–13,15,34–36]. In addition, a computer-vision approach is very flexible because it does not depend on the particular scenario, it is not specific, it does not consume much time and it is simple to set up [4].

Our algorithm is based on using a CNN model to detect the person’s skeleton into every frame, which is similar to other works (e.g., [25,26,39]). With respect to other works in the literature, we compute different features from the skeleton of an imaged person. In particular, in our approach, we represent the sequence of skeletons by a set of Gramian matrices, which result into a trajectory of points on the Riemannian manifold of positive semidefinite matrices. After that, we employ a Riemannian metric, a Gaussian Kernel and the DTW algorithm, in order to compute similarity scores between sequences. In the last step, we employ a linear SVM to classify between fall and non-fall events using similarity scores. Using the skeleton of the person has the clear advantage of overcoming the noise, which can occur by removing the background [23]. We also employ the Riemannian manifold in a different way with respect to that employed in other works (e.g., [23]). In addition, our algorithm does not depend on the person’s information, such as color. This differs from other works in the literature that cannot detect the silhouette of the person when the color of the person’s clothes is similar to the background [22,23,57]. In addition, our approach is able to detect the skeleton of the person in a video sequence that contains other moving objects, in contrast to methods that detect the silhouette of the person by removing the background (e.g., [21,52,54–56,60]).

Our approach only takes into consideration changes in the person’s skeleton during a video sequence, or a difference in person acceleration during a fall and non-fall real life event. In addition, our approach does not depend on the position of the camera. Furthermore, our algorithm shows a high classification rate on the URFD and Charfi datasets, as reported in Tables 2 and 4. The same tables also show our approach is competitive with respect to other state-of-the-art methods.

Limitations: Some problems still occur in our algorithm as in most of the computer-vision systems. Our algorithm aims to detect falls for a single person living alone at home, and it cannot manage multiple persons. In addition, our method detect sequences of daily activities as fall sequences, when the person’s acceleration is high. For this reason, it reports a specificity of 87% on Charfi and 93% on URFD. Furthermore, it is very difficult to detect the skeleton in dark environments using V2V-PoseNet.

From a more general point of view, fall detection research still suffers from some inherent limitations. The most evident one is related to the nature of the available data. The majority of fall detection datasets are small due to the number of participants. They also contain only a few simulated falls. For this reason, the validity of the test performed on such data is diminished and the reproducibility in real world scenarios needs to be proved. However, this seems to still be a difficulty that is problematic to remove. A further limitation of fall detection from simulated data is the inability to handle imbalanced datasets. In the real world, there are many more non-fall events than fall ones. Due to this, the accuracy is

biased toward correct detection of non-fall events rather than correct detection of falls. In addition, the majority of fall detection datasets do not take into consideration objects that an aged person can employ such as crutches. For works based on background removal, it is necessary to take into consideration crutches. In our case, the crutch does not cause an occlusion problem, because the skeleton of the person can be detected with or without crutches.

5. Conclusions

In this paper, we present an algorithm to detect fall events in video sequences by using the manifold of positive semidefinite matrices. Our method consists of four steps. In the first step, the skeleton of the imaged person is extracted from every frame of a sequence. The sequence of skeletons is then represented on the manifold of positive semidefinite matrices, during the second step. After that, in the third step, we compute the similarity scores between sequences using the DTW algorithm with a Riemannian metric. In the last step, an SVM classifier with a linear kernel is used to classify between fall and non-fall sequences. In the experiments, we demonstrated that our method achieves results that are competitive with state-of-the-art solutions on the same datasets. As future work, we aim to extend our approach to data captured by IR cameras. To make our model dynamic, we will recompute the similarity scores matrix for each new video sequence.

Author Contributions: Methodology, A.Y.A., R.O.H.T. and M.D.; software, A.Y.A.; data curation, A.Y.A.; writing—original draft preparation, A.Y.A.; writing—review and editing, A.Y.A., M.D., P.P., S.B. and Y.T.; supervision, R.O.H.T. and Y.T.; project administration, R.O.H.T.; and funding acquisition, R.O.H.T. All authors read and agreed to the published version of the manuscript.

Funding: This research was funded by ANGEL PROJECT: video surveillance of elderly people, CNRST and MESRSFC.

Acknowledgments: This work was supported by ANGEL PROJECT: video surveillance of elderly people. Financed by CNRST and MESRSFC.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DTW	Dynamic Warping Time
SVM	Support Vectors Machine
WHO	World Health Organization
ROC	Receiver Operating Characteristic
CNN	Convolutional Neural Network

References

1. United Nations, Department of Economic and Social Affairs, Population Division. *World Population Ageing*; 2019. Available online: <https://www.un.org/en/development/desa/population/publications/pdf/ageing/WorldPopulationAgeing2019-Highlights.pdf> (accessed on 29 June 2021).
2. World Health Organization. WHO Clinical Consortium on Healthy Ageing 2019: Report of Consortium Meeting Held, Geneva, Switzerland. 2019. Available online: <https://www.who.int/publications/i/item/9789240009752> (accessed on 29 June 2021).
3. Centers for Disease Control and Prevention. Important Facts about Falls. 2017. Available online: <https://www.cdc.gov/homeandrecreationalafety/falls/adultfalls.html> (accessed on 29 June 2021).
4. Mubashir, M.; Shao, L.; Seed, N.L. A survey on fall detection: Principles and approaches. *Neurocomputing* **2013**, *100*, 144–152. [CrossRef]
5. Xu, T.; Zhou, Y.; Zhu, J. New Advances and Challenges of Fall Detection Systems: A Survey. *Appl. Sci.* **2018**, *8*, 418. [CrossRef]
6. Igual, R.; Medrano, C.; Plaza, I. Challenges, issues and trends in fall detection systems. *BioMed. Eng. OnLine* **2013**, *12*, 66. [CrossRef]
7. Singh, K.; Rajput, A.; Sharma, S. Human Fall Detection Using Machine Learning Methods: A Survey. *Int. J. Math. Eng. Manag. Sci.* **2020**, *5*, 49–54. [CrossRef]

8. Iliev, I.T.; Tabakov, S.D.; Dotswinsky, A. Automatic fall detection of elderly living alone at home environment. *Glob. J. Med. Res.* **2011**, *11*, 161–180.
9. Wu, F.; Zhao, H.; Zhao, Y.; Zhong, H. Development of a wearable-sensor-based fall detection system. *Int. J. Telemed. Appl.* **2015**, *2015*, 576364. [[CrossRef](#)]
10. Shahiduzzaman, M. Fall detection by accelerometer and heart rate variability measurement. *Glob. J. Comput. Sci. Technol.* **2016**.
11. Yu, S.; Chen, H.; Brown, R.A. Hidden Markov model-based fall detection with motion sensor orientation calibration: A case for real-life home monitoring. *IEEE J. Biomed. Health Inform.* **2017**, *22*, 1847–1853. [[CrossRef](#)]
12. Zhao, S.; Li, W.; Niu, W.; Gravina, R.; Fortino, G. Recognition of human fall events based on single tri-axial gyroscope. In Proceedings of the 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), Zhuhai, China, 27–29 March 2018; pp. 1–6.
13. Chelli, A.; Pätzold, M. A Machine Learning Approach for Fall Detection and Daily Living Activity Recognition. *IEEE Access* **2019**, *7*, 38670–38687. [[CrossRef](#)]
14. Hussain, F.; Hussain, F.; ul Haq, M.E.; Azam, M. Activity-Aware Fall Detection and Recognition Based on Wearable Sensors. *IEEE Sens. J.* **2019**, *19*, 4528–4536. [[CrossRef](#)]
15. Musci, M.; Martini, D.; Blago, N.; Facchinetti, T.; Piastra, M. Online Fall Detection using Recurrent Neural Networks. *arXiv* **2018**, arXiv:1804.04976.
16. Geertsema, E.; Visser, G.; Viergever, M.; Kalitzin, S. Automated remote fall detection using impact features from video and audio. *J. Biomech.* **2019**, *88*, 25–32. [[CrossRef](#)]
17. Khan, M.S.; Yu, M.; Feng, P.; Wang, L.; Chambers, J. An unsupervised acoustic fall detection system using source separation for sound interference suppression. *Signal Process.* **2015**, *110*, 199–210. [[CrossRef](#)]
18. Nasution, A.H.; Emmanuel, S. Intelligent video surveillance for monitoring elderly in home environments. In Proceedings of the 2007 IEEE 9th Workshop on Multimedia Signal Processing, Chania, Greece, 1–3 October 2007; pp. 203–206.
19. Zhang, J.; Wu, C.; Wang, Y. Human Fall Detection Based on Body Posture Spatio-Temporal Evolution. *Sensors* **2020**, *20*, 946. [[CrossRef](#)]
20. Nogas, J.; Khan, S.S.; Mihailidis, A. DeepFall: Non-Invasive Fall Detection with Deep Spatio-Temporal Convolutional Autoencoders. *J. Healthc. Inform. Res.* **2018**, *4*, 50–70. [[CrossRef](#)]
21. Poonsri, A.; Chiracharit, W. Fall detection using Gaussian mixture model and principle component analysis. In Proceedings of the 2017 9th International Conference on Information Technology and Electrical Engineering (ICITEE), Phuket, Thailand, 12–13 October 2017; pp. 1–4.
22. Charfi, I.; Mitéran, J.; Dubois, J.; Atri, M.; Tourki, R. Definition and Performance Evaluation of a Robust SVM Based Fall Detection Solution. In Proceedings of the 2012 Eighth International Conference on Signal Image Technology and Internet Based Systems, Sorrento, Italy, 25–29 November 2012; pp. 218–224.
23. Yun, Y.; Gu, I.Y.H. Human fall detection via shape analysis on Riemannian manifolds with applications to elderly care. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, 27–30 October 2015, pp. 3280–3284.
24. Yun, Y.; Gu, I. 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. *Comput. Vis. Image Underst.* **2016**, *148*, 111–122. [[CrossRef](#)]
25. Chen, W.; Jiang, Z.; Guo, H.; Ni, X. Fall Detection Based on Key Points of Human-Skeleton Using OpenPose. *Symmetry* **2020**, *12*, 744. [[CrossRef](#)]
26. Yao, L.; Yang, W.; Huang, W. A fall detection method based on a joint motion map using double convolutional neural networks. *Multimed. Tools Appl.* **2020**, *1*–18. [[CrossRef](#)]
27. Kawatsu, C.; Li, J.; Chung, C.J. Development of a fall detection system with Microsoft Kinect. In *Robot Intelligence Technology and Applications 2012*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 623–630.
28. Alazrai, R.; Zmily, A.; Mowafi, Y. Fall detection for elderly using anatomical-plane-based representation. In Proceedings of the 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Chicago, IL, USA, 26–30 August 2014; pp. 5916–5919.
29. Pathak, D.; Bhosale, V.K. Fall Detection for Elderly People in Indoor Environment using Kinect Sensor. *International Journal of Science and Research* **2015**, *6*, 1956–1960.
30. Seredin, O.; Kopylov, A.; Huang, S.C.; Rodionov, D. A Skeleton Feature-based Fall Detection Using Microsoft Kinect V2 with one Class-Classifer Outlier Removal. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *4212*, 189–195. [[CrossRef](#)]
31. Abobakr, A.; Hossny, M.; Nahavandi, S. A Skeleton-Free Fall Detection System From Depth Images Using Random Decision Forest. *IEEE Syst. J.* **2018**, *12*, 2994–3005. [[CrossRef](#)]
32. Moon, G.; Chang, J.Y.; Lee, K.M. V2V-PoseNet: Voxel-to-Voxel Prediction Network for Accurate 3D Hand and Human Pose Estimation from a Single Depth Map. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5079–5088.
33. Szczapa, B.; Daoudi, M.; Berretti, S.; Bimbo, A.D.; Pala, P.; Massart, E. Fitting, Comparison, and Alignment of Trajectories on Positive Semi-Definite Matrices with Application to Action Recognition. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Korea, 27–28 October 2019; pp. 1241–1250.

34. Torti, E.; Fontanella, A.; Musci, M.; Blago, N.; Pau, D.; Leporati, F.; Piastra, M. Embedded Real-Time Fall Detection with Deep Learning on Wearable Devices. In Proceedings of the 2018 21st Euromicro Conference on Digital System Design (DSD), Prague, Czech Republic, 29–31 August 2018; pp. 405–412.
35. Nahian, M.J.A.; Ghosh, T.; Banna, M.H.A.; Aseeri, M.; Uddin, M.N.; Ahmed, M.R.; Mahmud, M.; Kaiser, M.S. Towards an Accelerometer-Based Elderly Fall Detection System Using Cross-Disciplinary Time Series Features. *IEEE Access* **2021**, *9*, 39413–39431. [[CrossRef](#)]
36. Kerdjij, O.; Ramzan, N.; Ghanem, K.A.; Amira, A.; Chouireb, F. Fall detection and human activity classification using wearable sensors and compressed sensing. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 349–361. [[CrossRef](#)]
37. Verlekar, T.T.; Soares, L.D.; Correia, P. Automatic Classification of Gait Impairments Using a Markerless 2D Video-Based System. *Sensors* **2018**, *18*, 2743. [[CrossRef](#)]
38. Jeong, S.; Kang, S.; Chun, I. Human-skeleton based Fall-Detection Method using LSTM for Manufacturing Industries. In Proceedings of the 2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), Jeju, Korea, 23–26 June 2019; pp. 1–4.
39. Alaoui, A.Y.; Fkihi, S.E.; Thami, R.O.H. Fall Detection for Elderly People Using the Variation of Key Points of Human Skeleton. *IEEE Access* **2019**, *7*, 154786–154795. [[CrossRef](#)]
40. Loureiro, J.; Correia, P. Using a Skeleton Gait Energy Image for Pathological Gait Classification. In Proceedings of the 2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020), Buenos Aires, Argentina, 16–20 November 2020; pp. 503–507.
41. Wu, S.; Khosla, Y.; Zhang, T. 3 D ShapeNets : A Deep Representation for Volumetric Shape Modeling. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
42. Massart, E.; Gousenbourger, P.Y.; Son, N.T.; Stykel, T.; Absil, P.A. Interpolation on the manifold of fixed-rank positive-semidefinite matrices for parametric model order reduction: preliminary results. In Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2019), Bruges, Belgium, 24–26 April 2019.
43. Meyer, G.; Bonnabel, S.; Sepulchre, R. Regression on Fixed-Rank Positive Semidefinite Matrices: A Riemannian Approach. *arXiv* **2011**, arXiv:1006.1288.
44. Vandereycken, B.; Absil, P.A.; Vandewalle, S. A Riemannian geometry with complete geodesics for the set of positive semidefinite matrices of fixed rank. *IMA J. Numer. Anal.* **2013**, *33*, 481–514. [[CrossRef](#)]
45. Bonnabel, S.; Sepulchre, R. Riemannian Metric and Geometric Mean for Positive Semidefinite Matrices of Fixed Rank. *SIAM J. Matrix Anal. Appl.* **2009**, *31*, 1055–1070. [[CrossRef](#)]
46. Massart, E.; Absil, P.A. Quotient Geometry with Simple Geodesics for the Manifold of Fixed-Rank Positive-Semidefinite Matrices. *SIAM J. Matrix Anal. Appl.* **2020**, *41*, 171–198. [[CrossRef](#)]
47. Journée, M.; Bach, F.R.; Absil, P.A.; Sepulchre, R. Low-Rank Optimization on the Cone of Positive Semidefinite Matrices. *SIAM J. Optim.* **2010**, *20*, 2327–2351. [[CrossRef](#)]
48. Massart, E.; Hendrickx, J.; Absil, P.A. Curvature of the Manifold of Fixed-Rank Positive-Semidefinite Matrices Endowed with the Bures-Wasserstein Metric. In Proceedings of the International Conference on Geometric Science of Information, Toulouse, France, 27–29 August 2019.
49. Kacem, A.; Daoudi, M.; Amor, B.B.; Berretti, S.; Alvarez-Paiva, J.C. A Novel Geometric Framework on Gram Matrix Trajectories for Human Behavior Understanding. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 1–14. [[CrossRef](#)] [[PubMed](#)]
50. Kacem, A.; Daoudi, M.; Amor, B.B.; Alvarez-Paiva, J.C. A Novel Space-Time Representation on the Positive Semidefinite Cone for Facial Expression Recognition. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 3199–3208.
51. Gudmundsson, S.; Runarsson, T.; Sigurdsson, S. Support vector machines and dynamic time warping for time series. In Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–8 June 2008; pp. 2772–2776.
52. Charfi, I.; Mitéran, J.; Dubois, J.; Atri, M.; Tourki, R. Optimised spatio-temporal descriptors for real-time fall detection: Comparison of SVM and Adaboost based classification. *J. Electron. Imaging* **2013**, *22*, 17. [[CrossRef](#)]
53. Kwolek, B.; Kepski, M. Human fall detection on embedded platform using depth maps and wireless accelerometer. *Comput. Methods Programs Biomed.* **2014**, *117*, 489–501. [[CrossRef](#)] [[PubMed](#)]
54. 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, PETRA'15, Corfu, Greece, 1–3 July 2015.
55. Chamle, M.; Gunale, K.; Warhade, K. Automated unusual event detection in video surveillance. In Proceedings of the 2016 International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 26–27 August 2016, Volume 2, pp. 1–4.
56. Alaoui, A.Y.; Elhassouny, A.; Thami, R.O.H.; Tairi, H. Human Fall Detection Using Von Mises Distribution and Motion Vectors of Interest Points. In Proceedings of the 2nd international Conference on Big Data, Cloud and Applications, BDCA'17, Tetouan Morocco, 29–30 March 2017.
57. Ali, S.F.; Khan, R.; Mahmood, A.; Hassan, M.T.; Jeon, M. Using Temporal Covariance of Motion and Geometric Features via Boosting for Human Fall Detection. *Sensors* **2018**, *18*, 1918. [[CrossRef](#)]

-
58. Kwolek, B.; Kepski, M. Improving fall detection by the use of depth sensor and accelerometer. *Neurocomputing* **2015**, *168*, 637–645. [[CrossRef](#)]
 59. Bourke, A.; O'Brien, J.; Lyons, G.M. Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. *Gait Posture* **2007**, *26*, 194–199. [[CrossRef](#)]
 60. Alaoui, A.Y.; Hassouny, A.E.; Thami, R.O.H.; Tairi, H. Video based human fall detection using von Mises distribution of motion vectors. In Proceedings of the 2017 Intelligent Systems and Computer Vision (ISCV), Fez, Morocco, 17–19 April 2017; pp. 1–5.