Abstract—Radar has a great potential to be one of the leading technologies to perform in-home monitoring of elderly. Radar signal returns corresponding to human gross-motor activities are nonstationary in nature. As such, time-frequency (TF) analysis plays a fundamental role in revealing constant and higher order velocity components of various parts of the human body under motion which are important for motion discrimination. In this paper, we consider radar for fall detection using TF-based deep learning approach. The proposed approach learns and captures the intricate properties of the TF signatures without human intervention and feeds the underlying features to the classifier. Experimental data is used to demonstrate the effectiveness of the proposed fall detection deep learning approach in comparison with the principal component analysis method and techniques incorporating manual selections of a few dominant features.

I. INTRODUCTION

The elderly population aged over 65 years is growing and their ratio to the population aged 20-64 will reach 35% in 2030 [1]. The worldwide population over 65 is projected to increase to one billion by 2030. An overwhelming majority of elderly exercise self-care at their own homes. Unfortunately, one out of three elderly will fall every year; the fall can result in injuries, reduced quality of life and, sadly, is considered one of the leading causes of death in the elderly population [2]. Most seniors are unable to get up by themselves after a fall and it was reported that, even without direct injuries, half of those who experienced an extended period of lying on the floor (more than an hour) passed away within six months after the incident. From an economical perspective, the high fall risk elderly will have to move to institutionalized care, which can cost in USA about $3,500 per month. Therefore, prompt fall detection saves lives, leads to timely interventions and most effective treatments, and reduces medical expenses. Further, it avoids major emotional and financial burdens on the elderly family.

Driven by the pressing need to detect and attend to a fall, elderly fall detection has become an active area of research and development and is identified by the US AARP as a major innovation opportunity to allow seniors to live independently [3]. There are competing methods for fall detection of which wearable devices, like accelerometers and push buttons, are most common [4]-[6]. The shortcomings of these devices are that they are intrusive, easily broken, and must be worn or carried. In addition, push-button devices are less suited for cognitively impaired users.

Smartphones are the latest technology to be utilized for fall detection. Smartphones have hardware that could be suitable for fall detection. Built-in inertial sensors, open source operating systems, state-of-the-art wireless connectivity and universal social acceptance can make smartphones an attractive alternative to conventional dedicated fall detection and prevention tools. However, the performance and usability of smartphones remain limited by the low quality of in-built sensors as well as the need to carry the smartphone in the same fixed position. The continuous usage of a smartphone for fall detection will certainly impact the battery life of the unit which in many cases is already a big concern.

Although in-home radar monitoring of elderly for fall detection, which is the subject of this paper, is still in its early stage of development, it carries great potential to be one of the leading technologies in the near future [7]. The attractive attributes of radar, with its proven technology, non-obstructive illumination, non-intrusive sensing, insensitivity to lighting conditions, privacy preservation, and safety, have brought electromagnetic waves to the forefront of indoor monitoring modalities in competition with cameras and wearable devices.

Human motions generate radar backscatters which represent highly non-stationary Doppler and microDoppler signals [8]-[13]. The characteristics of these signals are best revealed in the time-frequency (TF) domain. Typically, a small number of features are extracted from the TF signature of the radar data and applied to a classifier [14]-[17]. However, two issues arise when using the above approach, namely: a) feature extractions can be a tedious task involving tuning of parameters and thresholds, and b) the features can exhibit large variances, depending on the elderly being monitored and deviations from nominal bio-mechanical and kinematics forces. For example, a situation may arise where the boundaries of the Doppler signatures of falling and sitting motions appear similar to the naked eye and also to the classifiers, leading to considerable false alarm rates. One way to overcome these issues is to have an approach which learns and captures the detailed and intricate properties of the TF signature and feeds the learnt underlying features to the classifier. This can be achieved via deep learning.

In this paper, we propose an approach based on deep learning for fall detection using radar. This approach is based on the use of a deep neural network (DNN), which consists of stacked auto-encoders and a softmax layer output. The TF power distributions are cast as images and used as input to the DNN. The task of the auto-encoders is to provide a sparse representation of the TF data, i.e., to learn the most prominent features. These features are applied to the softmax layer leading to the classified data. We provide experimental results which demonstrate improved classification rates of the deep learning approach over existing methods where the features are pre-defined and manually selected. Additionally, we show that the proposed approach attains superior results compared to the commonly used principal component analysis (PCA) classification approach [12], [18].

The remainder of the paper is organized as follows. Section
II briefly reviews the conventional fall detection approach based on TF signatures, while Section III describes the deep learning approach. Classification performance of the deep learning approach is compared and contrasted with the conventional and PCA approaches using experimental data in Section IV. Conclusions are drawn in Section V.

II. CONVENTIONAL FALL DETECTION APPROACH

Consider a monostatic continuous-wave (CW) radar for sensing the human motion. A human is a spatially extended target, and, as such, can be modeled as a collection of point scatterers. Therefore, the corresponding radar return \( x(t) \) is the integration of individual point scatterer returns over the target region, and the corresponding Doppler signature is the superposition of all component Doppler frequencies. The dynamics of human motion, in general, generate time-varying Doppler frequencies, and their exact signatures depend on the target shape and motion patterns.

Since the Doppler radar return, \( x(t) \), from the human is nonstationary, TF analysis is a natural tool that reveals the local signal behavior and depicts its time-varying Doppler signatures with enhanced energy concentration [19]. A number of low- and high-resolution methods are available to perform TF analysis of the Doppler signatures [20]-[27], of which short-time Fourier transform (STFT) is a commonly used technique. The spectrogram \( S(t, f) \), which shows how the signal power varies with the time index \( t \) and frequency \( f \), is obtained by computing the squared magnitude of the STFT of the data \( x(t) \) with a window \( h(t) \), expressed as

\[
S(t, f) = \sum_{m=\infty}^{\infty} h(m)x(t-m)\exp(-j2\pi fm)^2. \tag{1}
\]

The window function \( h(t) \) trades off the time and frequency resolutions. A larger window length may degrade the time resolution whereas a smaller window length may compromise the frequency resolution. Fig. 1 depicts the spectrograms of the Doppler signatures of four different human motions, namely, falling, sitting, bending to pick an object and straightening, and walking. A Hamming window of length 255 is used for the computation of the spectrograms. The results are based on experimental data collected in the Radar Imaging Lab, Villanova University, using a 6 GHz CW radar.

For fall detection based on STFT, three features, namely, extreme frequency magnitude, extreme frequency ratio, and time-span of event, have been typically extracted from the spectrogram [15], [17]. The extreme frequency magnitude is defined as

\[
F = \max(f_{+\max}, -f_{-\min}), \tag{2}
\]

where \( f_{+\max} \) and \( f_{-\min} \), respectively, denote the maximum frequency in the positive frequency range and the minimum frequency in the negative frequency range. Critical falls often exhibit a high extreme frequency magnitude when compared to other types of observed motions. The extreme frequency ratio is defined as

\[
R = \max(|f_{+\max}/f_{-\min}|, |f_{-\min}/f_{+\max}|). \tag{3}
\]

For falls, due to the translational motion of the entire body, high energy spectrogram is concentrated in either the positive or negative frequencies, resulting in a high extreme frequency ratio. On the other hand, other types of motions, such as bending and straightening, often demonstrate high energy content in both the positive and negative frequency bands because different body parts undergo different motion patterns, thereby corresponding to a low extreme frequency ratio. Finally, the time-span of the event describes the length of time, in milliseconds, between the start and the end of an event,

\[
L = t_{\text{extrm}} - t_{\text{begin}}, \tag{4}
\]

where \( t_{\text{extrm}} \) denotes the time where the extreme frequency occurs and \( t_{\text{begin}} \) denotes the initiation time of the event. The latter is determined by the time when the magnitude of the frequency content of a signal passes a specific threshold.

Once the aforementioned three features are extracted, a classification algorithm is applied to classify the activity. A variety of classifiers have been employed for fall detection [16], [14], with the support vector machine (SVM) being the most commonly used classifier. Note that the choice of employed features has been determined to have a greater impact on the classification performance than the specific classifier applied (see [28] and references therein).

In view of the feature definitions in (2)-(4), a careful observation of the TF signatures in Fig. 1 reveals that these few features do not uniquely define the respective motion and fail to capture the intricate details of each motion signature, thereby missing out on crucial differentiating properties. As such, a deep learning based approach that can learn and capture important and key details of the TF signal representation is expected to be better suited to the underlying problem.

III. FALL DETECTION BASED ON DEEP LEARNING

In this section, we present a fall detection scheme based on deep learning. The block diagram of the procedure is depicted in Fig. 2 and the various stages are described below.

A. Preprocessing

When using deep learning algorithms, it is typical for some form of data preprocessing to occur [29]. This usually involves denoising and data normalization. For the application at hand, the preprocessing stage operates on the TF signatures obtained using the spectrogram. Denoising is performed by setting a threshold after estimating the noise in the spectrogram [15]. In addition, the DC component of the TF signature is removed, as it does not contain any useful information for motion classification. The processed spectrogram is then converted to a gray-scale image that represents the input to the DNN, which consists of stacked auto-encoders and a softmax regression classifier.

B. Stacked Auto-Encoders

An auto-encoder is defined as a neural network which attempts to reconstruct its input at its output [29]. In its simplest form, it contains only one hidden layer. The weights and biases of neurons are learned such that the reconstruction error is minimized. The learning is performed in an unsupervised manner, i.e., no labeling of the data is provided. If the number of hidden neurons is smaller than the number of input
neurons, then the auto-encoder attempts to learn the sparse representation of the input data.

The weights $w$ and biases $b$ in the sparse auto-encoder are obtained by minimizing the following cost function,

$$J(w, b) = E(w, b) + \beta D_{KL}(\rho, \hat{\rho}).$$

(5)

Here, the first term $E(w, b)$ represents the error between the input data $u$ and the auto-encoder output $y$. In order to avoid overfitting, a regularization term is added which prevents weights from assuming high values. Thus, $E(w, b)$ is defined as

$$E(w, b) = \frac{1}{2} \| y(u, w, b) - u \|^2 + \lambda \| w \|^2.$$  

(6)

where $\lambda$ represents the regularization parameter. The second term in the cost function (5) is responsible for obtaining sparse representation. $D_{KL}(\rho, \hat{\rho})$ represents the Kullback-Leibler divergence between $\rho$, a sparsity parameter, and $\hat{\rho}$, which is an average output of the hidden neurons. The goal is to minimize the difference between $\hat{\rho}$ and $\rho$; the latter typically assumes a small value. The parameter $\beta$ in (5) determines the importance of two terms with respect to each other.

By computing the sparse representation, we are essentially extracting the most prominent features. Since images of human motions contain significant amount of useful information, it is prudent to extract this information in several layers where each layer represents a different concept of the input data. For example, one layer can learn the edges, while the next layer can learn the shapes which contain these edges. This manner of learning input representation in multiple levels can be achieved using stacked auto-encoders, where output of one auto-encoder is input to the next one. In this paper, the feature extraction is performed using two sparse auto-encoders; the choice of the number of layers for the underlying problem has been empirically determined.

C. Softmax Regression Classifier

The output $z$ of the stacked auto-encoders is input to a softmax regression classifier [29]. The output of the classifier is defined as an $L$-dimensional vector, where $L$ denotes the number of classes which we are trying to distinguish. The $l$th element of the output vector contains the estimated probability $p_l$ for the event that the data $z$ belongs to the class label $y_l$. The element with the highest probability determines the class...
to which the test data is assigned. The probability $p_l$, is defined as
\[
p_l = \frac{1}{\sum_{l=1}^{L} e^{\theta_l z}}, \quad l = 0, 1, \ldots, L - 1, \tag{7}
\]
where the parameter $\theta_l$ is determined by minimizing an objective function, which is based on the indicator function $1\{\cdot\}$ as
\[
J(\theta) = \sum_{l} 1\{y = l\} \log \frac{e^{\theta_l z}}{\sum_{l=1}^{L} e^{\theta_l z}}. \tag{8}
\]
Typically, a regularization term is added in (8) to prevent overfitting.

### IV. Experimental Results

In this section, we demonstrate the effectiveness of the deep learning based fall detector. Measurements were made using a monostatic CW radar in the Radar Imaging Lab at Villanova University. A vertically polarized horn antenna (BAE Systems, Model H-1479) with an operational frequency range of 1-12.4 GHz and 3-dB beamwidth of 45° was used as the transceiver. The feed point of the antenna was positioned 1 m above the floor. Agilent’s E5071B RF network analyzer was used for signal generation and measurement of radar returns. A carrier frequency of 6 GHz was employed with the transmit power set as 3 dBm. The network analyzer was externally triggered to obtain a 1 kHz sampling rate. Data were collected for four different motion patterns (falling, sitting, bending and straightening, walking) using several test subjects, both male and female. Two different variations of each motion were measured, one being a normal version and the other corresponding to a higher speed form of the same motion in order to study the impact of such variations on the classification performance. A total of 120 experiments were recorded. The record time of each experiment is 20 seconds, resulting in a total of 20,000 data samples per experiment.

One-half of the recorded data, i.e., 60 signals, are used for training, whereas the remaining 60 are employed in the testing phase. For each experiment, the TF signature corresponding to a time span of 3 s containing the motion is preprocessed and the resulting gray-scale image consists of $76 \times 70 = 5320$ pixels. The image is used as input to the first auto-encoder. The number of outputs for the first auto-encoder is set to 300. In other words, the network is attempting to compress 5320 coefficients into 300 outputs. These 300 outputs are input to the second encoder, which further compresses the data by a factor of 2, resulting in 150 outputs. The final stage is the softmax regression classifier, which assigns the test data to one of four possible classes, namely, falling, sitting, walking, and bending. The confusion matrix for the DNN based classifier is given in Table I. We observe that all walking signatures have been successfully detected. For the case of falls, we have one missed detection and three false alarms. More specifically, some of the sitting signatures are misclassified as falls. This is expected since a few subjects sat quickly as part of the experiments which led to the TF signature of sitting resembling that of a fall.

For comparison, we also present results obtained using the conventional approach described in Section II. Three extracted features, namely, the extreme frequency magnitude, extreme frequency ratio, and time interval of event, are fed into an SVM classifier. The corresponding confusion matrix is provided in Table II. Comparing Tables I and II, we note that the deep learning approach provides better fall detection with fewer missed detections and false alarms. It should be noted that in the conventional approach, special care was exercised for the tuning of various parameters and thresholds in order to find the best match for the considered data. Despite the extensive amount of efforts expended for parameter tuning, an overall success rate of 78% was achieved for the conventional approach in contrast to 87% for the deep learning approach. Note that the success rate is defined as the total number of true positives divided by the size of the testing set.

Since deep learning can be regarded as a generalization of the PCA [30], we also provide classification results using the most prominent principal components. PCA-based motion classification comprises the following steps. First, each pre-processed gray-scale spectrogram image in the training set is vectorized and stored as a column of the ‘training matrix’. Next, the average of the vectorized training images is subtracted from each column of the training matrix. The resulting training matrix is used for generating the eigenvalues and the eigenvectors (eigen images). After performing the eigendecomposition and choosing the eigen images corresponding to the dominant eigenvalues, we project the training set onto the space spanned by the selected eigen images. These projections are used in the classification process. When the test data arrives, it is projected onto the eigen space and the resulting projection is compared with all projections obtained for the training images. Minimum Euclidean distance determines the class with the closest match to the observed test motion. The normalized eigenvalues corresponding to the training set with 60 images are plotted in Fig. 3, which shows that there are four dominant components. The corresponding eigen images are depicted in Fig. 4. Table III provides the confusion matrix when the first four eigen images are used in the PCA-based scheme, which achieves a success rate of 73%. By employing all 60 components, we obtain 83% success rate, which is still lower than 87% for the deep learning detector.

In order to examine the impact of the preprocessing stage on the performance of the deep learning based detector, we provide the ‘raw’ spectrograms (without any preprocessing) as input to the DNN. The corresponding confusion matrix is depicted in Table IV. We observe that there is a high number...
Table III. Confusion Matrix for the PCA-Based Approach. Numbers correspond to the case when 4 (60) components are used.

<table>
<thead>
<tr>
<th>Predicted/Actual Class</th>
<th>Walk</th>
<th>Fall</th>
<th>Sit</th>
<th>Bend</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walk</td>
<td>15(15)</td>
<td>0(0)</td>
<td>0(0)</td>
<td>0(0)</td>
</tr>
<tr>
<td>Fall</td>
<td>0(0)</td>
<td>10(11)</td>
<td>4(2)</td>
<td>1(0)</td>
</tr>
<tr>
<td>Sit</td>
<td>0(0)</td>
<td>5(4)</td>
<td>10(12)</td>
<td>5(3)</td>
</tr>
<tr>
<td>Bend</td>
<td>0(0)</td>
<td>0(0)</td>
<td>1(1)</td>
<td>9(12)</td>
</tr>
</tbody>
</table>

![Normalized eigenvalues corresponding to the 60 training images.](image)

**Fig. 3.** Normalized eigenvalues corresponding to the 60 training images.

Table IV. Confusion Matrix for the case when raw spectrograms are used as the input.

<table>
<thead>
<tr>
<th>Predicted Class/Actual Class</th>
<th>Walk</th>
<th>Fall</th>
<th>Sit</th>
<th>Bend</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walk</td>
<td>15</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Fall</td>
<td>0</td>
<td>12</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>Sit</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bend</td>
<td>0</td>
<td>3</td>
<td>5</td>
<td>8</td>
</tr>
</tbody>
</table>

V. CONCLUSION

In this paper, we have applied DNN for radar-based fall motion detection. The employed DNN consisted of two stacked auto-encoders and a softmax regression classifier. Four human motions were considered, namely, walking, falling, bending/straightening, and sitting. Experimental results were provided, which demonstrated the superiority of the DNN based approach over the conventional and PCA based methods in discriminating between the different considered human motions.

REFERENCES

Fig. 4. Eigen images corresponding to the first four principal components.

Fig. 5. Raw (top row) and preprocessed (bottom row) spectrograms: (a) Falling, (b) Sitting, (c) Bending and straightening, (d) Walking.


