Spatio-Temporal Adversarial Learning for Detecting Unseen Falls

Shehroz S. Khan · Jacob Nogas · Alex Mihailidis

Received: date / Accepted: date

Abstract Fall detection is an important problem from both the health and machine learning perspective. A fall can lead to severe injuries, long term impairments or even death in some cases. In terms of machine learning, it presents a severely class imbalance problem with very few or no training data for falls owing to the fact that falls occur rarely. In this paper, we take an alternate philosophy to detect falls in the absence of their training data, by training the classifier on only the normal activities (that are available in abundance) and identifying a fall as an anomaly. To realize such a classifier, we use an adversarial learning framework, which comprises of a spatio-temporal autoencoder for reconstructing input video frames and a spatio-temporal convolution network to discriminate them against original video frames. 3D convolutions are used to learn spatial and temporal features from the input video frames. The adversarial learning of the spatio-temporal autoencoder will enable reconstructing the normal activities of daily living efficiently; thus, rendering detecting unseen falls plausible within this framework. We tested the performance of the proposed framework on camera sensing modalities that may preserve an individual’s privacy (fully or partially), such as thermal and depth camera. Our results on three publicly available datasets show that the proposed spatio-temporal adversarial framework performed better than other frame based (or spatial) adversarial learning methods.

Keywords Falls, Spatio-Temporal, Adversarial Learning, Autoencoder, Thermal Camera, Depth Camera

1 Introduction

Falls can cause severe injuries to people resulting in permanent or partial disability, huge health care costs and development of negative social and psychological problems [17]. This
constitutes a strong motivation to detect falls; however, due to their rarity of occurrence, traditional supervised machine learning classifiers are difficult to use for this task [6]. In many cases, there may be very few or no falls data available during training because collecting falls data is very challenging and can put people’s life in danger. On the other hand, normal activities of daily living (ADL) performed by people are abundantly available and easier to collect. Therefore, we propose to detect falls in a one-class classification (OCC) framework [7] that enables a classifier to learn only on the normal ADL and be able to detect an unseen fall during testing (as they may not be present during training).

Learning one-class classifiers from video sequences of normal ADL to detect falls as an anomaly is a challenging task [12]. The idea of detecting a fall as an anomaly from videos is different and difficult in comparison to finding general anomalies in videos. The general anomaly detection methods seek to find any irregularity or deviation from the normal behaviour and raises a flag. Applying similar analogy to fall detection may result in too many false alarms and may render the method ineffective for practical purposes [4]. A fall is a spatio-temporal event; thus, the decision should be taken on a sequence of frames rather than individual frames because that can also increase the number of false alarms. Another challenge in video based fall detection is to preserve the privacy of the person, which traditional RGB cameras cannot provide. Thus, detecting falls in videos without explicitly knowing a person’s identity is important from the real world usability of such systems.

The learning paradigm using generative adversarial networks (GAN) presents a unique opportunity to not only mimic normal behaviour through the generator but also to effectively discriminate it from anomalies [16]. In this paper, we design a new spatio-temporal adversarial learning framework, which consists of a spatio-temporal convolutional autoencoder (3DCAE) to reconstruct a sequence of input video frames and a spatio-temporal convolutional neural network (3DCNN) as a classifier to discriminate them from the original sequence of video frames. The spatio-temporal architecture of the adversarial framework consists of 3D convolutional layers that will extract both spatial and temporal features from the video frames that will result in a robust system to learn normal ADL from the video sequences. After the training is completed, the 3DCAE would be able to reconstruct ADL sequences efficiently and the 3DCNN would be able to differentiate between real and reconstructed ADL sequences. Therefore, this framework would be able to identify unseen falls with high accuracy. To achieve that, the reconstruction error of the 3DCAE or the probability output of the 3DCNN or their combination can be used as an anomaly score to identify unseen falls during testing. To the best of our knowledge, this is the first spatio-temporal adversarial framework for anomaly detection in video sequences, in particular for detecting unseen falls. We use two computer vision sensing modalities, thermal and depth cameras, to test our method. Both these sensing modalities can partially or fully mask the facial identity of the person; thus, they are more promising to be used in a home-setting. We also implemented two spatial variations of adversarial learning framework with (i) an autoencoder to reconstruct input frames and a deep neural network as a discriminator, and (ii) a convolutional autoencoder to reconstruct input frames and a CNN as a discriminator (similar to the work of [15]). The input to both of these methods is a frame from the video, whereas the input to our proposed method is a sequence of video frames. Our results on three publicly available fall detection datasets captured using thermal and depth cameras show superior performance of the spatio-temporal adversarial learning framework in detecting unseen falls in comparison to these spatial adversarial approaches.

The paper is organized as follows. In Section 2, we present literature review on using adversarial techniques for anomaly detection in images and videos. In Section 3, we introduce the proposed temporal adversarial learning framework. Section 4 presents various
anomaly scores to detect unseen falls. The experiments and results are described in Section 5, followed by conclusions and pointers to future research in Section 6.

2 Related Work

In this paper, we detect falls in an OCC framework. To the best of our knowledge, fall detection has not been addressed using an adversarial learning framework; therefore, we present literature review on techniques that use adversarial learning for general anomaly detection in images and videos.

One of the earliest work to detect anomalies using GAN framework is presented by Schlegl et al. [16] to find anomalies in imaging data as candidates for markers, called as AnoGan. The generator of their GAN is equivalent to a multi-layered convolutional decoder that samples input from uniformly distributed noise. The discriminator is a standard CNN that maps 2D images to a single value that can be interpreted as a probability whether the input to it is a real image or is produced by the generator. They use the combination of residual and discrimination losses as an anomaly score, such that a large score means an anomalous image. Eide [3] applied generative adversarial learning to find anomalies in hyper-spectral remote sensing images. Their generator is based on ResNet, which maps low-dimensional input to a higher dimension image; thus, works as a convolutional decoder. The discriminator has similar design but works in opposite direction. They modify the reconstruction cost of the generator by adding a term for the norm of generated input. The modified reconstruction cost penalizes reconstructions from unlikely inputs more heavily. However, adding this term is not found to be helpful as the generator is unable to reconstruct anomalies even without any penalty term. Ravanbaksh et al. [14] present the use of GAN for anomaly detection in crowded scenes. They train two conditional GANs; one each for generating optical-flow from frames and the other generating frames from optical-flow using input image and noise vector as inputs. The conditional discriminator takes either of the generated images and compares against the real image to produce a probability that both of its input images come from the real data. However, this method may not work well with occluded scenes and it may be difficult to estimate the optical flow map. Lawson et al. [8] present the use of deep convolutional GAN for finding anomalies in autonomous robot patrol view. Their method first learns the model for normal scene using GAN and then use the features learned to find anomalies in the environment. More specifically, they compare the difference between the bottleneck features extracted with real images and reconstructed images and use it as a measure for finding anomalies. Yu et al. [20] present a GAN in an OCC framework for generating negative data for observed classes that can make the recognition of unseen classes easy using supervised methods. They present an objective function that can generate boundary samples that do not belong to each of the observed classes, but should be different from each other and closer to their respective classes. Their approach is shown to be more effective than several state-of-the-art OCC methods on different datasets.

Yarlagadda et al. [19] present the use of GAN for satellite image forgery detection and localization. The generator in their structure is an autoencoder and the discriminator is a CNN. The adversarially trained autoencoder encodes the image patches into low dimensional features, which are then used to train a one-class SVM to detect forged patches. Sabokrou et al. [15] present an end-to-end OCC method that uses adversarial learning. The generator of their network is a convolutional autoencoder, which reconstructs the input with added noise. The discriminator is a typical CNN that takes reconstructed and real input and gives a likelihood estimate of the target score. After the adversarial training, the discrimi-
nator can be used to detect anomalies. They also show that applying discriminator on the reconstructed images can provide better separation; hence, better performance. Their results on MNIST, Caltech-256 and UCSD Ped2 (video) datasets show the viability of learning one-class classifiers in an adversarial manner. Lee et al. [9] present a spatio-temporal adversarial learning framework for anomaly detection in videos. Their framework consists of a spatio-temporal generator and discriminator. The network operates on a sequence of $N + 1$ video frames. The generator takes as input the first and last $\frac{N}{2}$ frames and then generates the missing $\frac{N}{2} + 1^{th}$ frame. This middle frame is generated by a bi-directional convolutional LSTM network. The discriminator consists of a 3DCAE that takes a sequence of $N + 1$ frames as input, which has one generated frame and rest are original frames. The discriminator then tries to recognize this sequence as fake, while the generator must improve to generator for the middle frame in order to fool the discriminator. A potential issue with such an approach is that the discriminator is given a very difficult task to only detect one frame in a sequence; and conversely the generator is given an easy task.

The spatio-temporal adversarial learning method to detect unseen falls presented in this paper extends the work of Sabokrou et al. [15] from single image to a sequence of images (video) by learning spatio-temporal features. The proposed frame also differs from the work of Lee et al. [9] in that it uses a 3DCAE instead of the bi-directional convolutional LSTM or 2D CAE. The work of Nogas et al. [13] suggests that training LSTM based autoencoders can be very slower in comparison to 3DCAE. Our 3DCAE reconstructs the whole sequence of frames given an input sequence of frames instead of producing only one frame and is fed to the 3DCNN discriminator. This way the discriminator is presented with a fully reconstructed sequence of frame, rather than one frame in a sequence to decide if its real or reconstructed sequence of frames. In the next section, we describe the various components of the proposed spatio-temporal adversarial framework.

3 Spatio-Temporal Adversarial Learning

We propose to use spatio-temporal adversarial learning for identifying unseen falls. Our framework consists of: (i) training a 3DCAE to reconstruct a sequence of normal ADL video frames and, (ii) a 3DCNN to discriminate them with the original sequences of video frames of normal ADL. Both of these components perform 3D Convolution operations. A 3D convolutional layer is defined as follows: the value $v$ at position $(x, y, z)$ of the $j^{th}$ feature map in the $i^{th}$ 3D convolution layer, with bias $b_{ij}$, is given by the equation [5]

$$v_{ij}^{xyz} = f(\sum_{m} \sum_{p=0}^{P_{i}-1} \sum_{q=0}^{Q_{i}-1} \sum_{s=0}^{S_{i}-1} w_{ijm}^{pqs}v_{(i-1)jm}^{(x+p)(y+q)(z+s)} + b_{ij})$$

(1)

where $P_{i}$, $Q_{i}$, $S_{i}$ are the vertical (spatial), horizontal (spatial), and temporal extent of the filter cube $w_{ij}$ in the $i^{th}$ layer. The set of feature maps from the $(i-1)^{th}$ layer are indexed by $m$, and $w_{ijm}^{pqs}$ is the value of the filter cube at position $pqs$ connected to the $m^{th}$ feature map in the previous layer. Multiple filter cubes will output multiple feature maps. Next, we describe the 3DCAE and 3DCNN that will be used in the proposed adversarial learning framework to detect unseen falls.
### 3.1 3DCAE

The autoencoder in our setting is a 3DCAE, the input to it, \( I \), comprises of a continuous sequence of \( t = 1, \ldots, T \) frames, called a window. These windows of length \( T \) are generated by applying a temporal sliding window to input video frames, with padding (or not) and stride (the amount of frames shifted from one window to the next). The input \( I \) is encoded by a sequence of 3D convolution layers. The first 3D convolution layer uses 3D convolutions with stride of \( 1 \times 2 \times 2 \), and padding, and the rest use stride of \( 2 \times 2 \times 2 \), and padding. This means that each dimension (temporal depth, height, and width) is reduced by a factor of 2 with every 3D convolution layer except the first, which reduces only the spatial dimension, thus allowing for a deeper architecture without collapsing the temporal dimension completely. Decoding operates as encoding but in reverse, using 3D deconvolution layers. The final deconvolution layer combines feature maps into the decoded reconstruction. This final layer uses stride \( 1 \times 1 \times 1 \) and padding. For hidden layers, the activation function \( f \) is set to ReLU. We use \( P_i = Q_i = 3 \), and \( S_i = 5 \), for all convolutional and deconvolutional layers, as these values were found to produce the best results across all data sets. Table 1 shows the configuration of the 3DCAE used in our spatio-temporal adversarial framework. The output of the 3DCAE is fed to the 3D discriminator (along with actual input). Batch normalization is used in all the layers of the 3DCAE except for the final layer.

| Input          | (8, 64, 64, 1) |
|----------------|---------------|
| **Encoder**    |               |
| 3D Convolution | (8, 64, 64, 16)|
| 3D Convolution | (8, 32, 32, 8) |
| 3D Convolution | (4, 16, 16, 8) |
| 3D Convolution | (2, 8, 8, 8)  |
| **Decoder**    |               |
| 3D Deconvolution | (4, 16, 16, 8)|
| 3D Deconvolution | (8, 32, 32, 8)|
| 3D Deconvolution | (8, 64, 64, 16)|
| 3D Convolution | (8, 64, 64, 1) |

Table 1: Configuration of the 3D Generator

### 3.2 3D Discriminator

The discriminator in our setting is a 3DCNN, whose architecture is kept the same as the encoding configuration of the 3DCAE followed by a fully connected layer of one neuron at the end with a sigmoid function to output a probability of whether a sequence of frames is original or reconstructed. Batch normalization is used in all layers of the 3D discriminator except for the input layer. LeakyRelu activation is set in all hidden layers, with negative slope coefficient set to 0.2.

It is to be noted that during the training phase, only the video sequences of normal ADL are presented to the 3DCAE and 3DCNN, whereas during testing phase video sequences may contain both normal ADL and fall frames.
3.3 Adversarial Learning

As discussed previously, the proposed adversarial framework consists of two components; a 3DCAE and a 3DCNN as a discriminator. Figure 1 shows the setup of the overall framework, where the autoencoder and discriminator are trained in an adversarial setting. The 3DCAE (represented as \( R \)) takes the input sequence \( I \) of window size \( T \) of normal ADL, and reconstructs the sequence, \( O \), which is then fed to fool 3DCNN (represented as \( D \)) that it is an original input and not the reconstructed sequence. However, \( D \) will have access to the original input sequence \( I \) and may easily identify the reconstructed sequence as not the original input sequence. Then, the two components play an adversarial game, which after completion of training should enable \( R \) to reconstruct input video sequences with minimum reconstruction error to successfully fool \( D \). This means that \( R \) should be able to reconstruct output sequence very similar to the input sequence. In other words, the spatio-temporal autoencoder would have learned the concept of normal ADL after successful completion of the training. This further implies that any sequence with anomaly (e.g. fall) would be reconstructed with high reconstruction error. At the same time, the discriminator would have become an expert to identify between the badly reconstructed sequences and the input sequences.

![Block diagram of spatio-temporal adversarial framework to detect unseen falls.](image)

In our setting, \( R \) maps \( I \) to \( O \) using the distribution of the target class \( p \), i.e.

\[
O = I \sim p
\]  
(2)

However, \( D \) has access to input samples and is exposed to \( p \). Therefore, \( D \) can explicitly decide if \( R(O) \) comes from \( p \) or not. The objective function to jointly learn \( R \) and \( D \) can be written as:

\[
\min_{R} \max_{D} \left( \mathbb{E}_{I \sim p}[\log(D(I))] + \mathbb{E}_{O \sim p}[\log(1 - D(R(O)))] \right)
\]  
(3)

To train the model, we need to calculate the (i) loss due to the 3DCAE \( (L_{R}) \), and (ii) loss due to both 3DCAE and the 3DCNN \( (L_{R+D}) \). The 3DCAE loss is simply the reconstruction error between the \( j^{th} \) frame of \( I_i \) and \( O_i \), and can be written as

\[
L_{R} = \|I_{i,j} - O_{i,j}\|_2^2
\]  
(4)
Thus, the total loss function to minimize can be written as:

\[ \mathcal{L} = \mathcal{L}_R + \lambda \mathcal{L}_{R + D} \]  

(5)

where \( \lambda \) is a positive number that controls the relative importance of both the loss terms.

For comparison purposes, we implement two other variants of autoencoders to detect unseen falls that are trained as per the proposed adversarial framework. The first variant uses a deep autoencoder and a multi-layer feed forward network as the discriminator, we call it as DAE-AN. The configuration of the discriminator is the same as the encoder of the deep autoencoder. This method will learn global features from the video sequences to successfully reconstruct ADL. The second variant uses a convolutional autoencoder (CAE) and a convolutional feed-forward network as a discriminator, we call it as CAE-AN. The configuration of the discriminator, in this case, is the same as the encoder of the CAE (this framework is analogous to the work of Sabokrou et al. [15]). This method will learn localized spatial features. The structure of the encoder and decoder for DAE-AN and CAE-AN are shown in Tables 2 and 3. It is to be noted that the input to the DAE-AN and CAE-AN is a frame from the video, whereas the input to the proposed spatio-temporal adversarial learning method is a window comprising of a sequence of \( T \) frames, as shown in Figure 1. Therefore, the proposed method will learn both spatial and temporal features when the training is successfully completed.

| Input | (64, 64, 1) |
|-------|-------------|
| Encoder | Fully Connected - (4096) |
|        | Fully Connected - (1500) |
|        | Fully Connected - (1000) |
|        | Fully Connected - (500)  |
| Decoder| Fully Connected - (1000) |
|        | Fully Connected - (1500) |
|        | Fully Connected - (4096) |
|        | Fully Connected - (64, 64, 1) |

Table 2: Configuration of the DAE-AN. The values inside the parenthesis for fully connected layers are the number of neurons.

| Input | (64, 64, 1) |
|-------|-------------|
| Encoder | 2D Convolution - (64, 64, 16) |
|        | 2D Convolution - (32, 32, 16) |
|        | 2D Convolution - (16, 16, 8)  |
|        | 2D Convolution - (8, 8, 8)    |
| Decoder| 2D Deconvolution - (16, 16, 8) |
|        | 2D Deconvolution - (32, 32, 8) |
|        | 2D Deconvolution - (64, 64, 16) |
|        | 2D Deconvolution - (64, 64, 1) |

Table 3: Configuration of the CAE-AN. The values inside the parenthesis are the size of the convolution filters.
4 Detecting Unseen Falls

The spatio-temporal framework is trained in an adversarial manner on only normal ADL and an unseen fall is detected as an anomaly during testing. The strategy to detect unseen falls is shown in Figure 2 (derived from [13]). All the frames in the video, $F_{r_i}$, are broken down into windows of frames of length, $T = 8$, with stride=1. For the $i^{th}$ window $I_i$, the 3DCAE outputs a reconstruction of this window, $O_i$. The reconstruction error ($R_{i,j}$) between the $j^{th}$ frame of $I_i$ and $O_i$ can be calculated as (similar to Equation 4)

$$R_{i,j} = \| I_{i,j} - O_{i,j} \|^2_2 \quad (6)$$

There are two ways to detect unseen falls, (i) at the frame level, or (ii) at the window level, which are described next.

**Frame Level Anomaly**: In the frame level anomaly method, the reconstruction error ($R_{i,j}$) (obtained from the 3DCAE) is computed for every frame $j$ across different windows. The average ($C^j_{\mu}$) and standard deviation ($C^j_{\sigma}$) of a frame $j$ across different windows are used as an anomaly score as follows [13]:

$$C^j_{\mu} = \begin{cases} \frac{1}{j} \sum_{i=1}^{j} R_{i,j} & j < T \\ \frac{1}{T} \sum_{i=1}^{T} R_{i,j} & j \geq T \end{cases} \quad (7)$$

$$C^j_{\sigma} = \begin{cases} \sqrt{\frac{1}{j} \sum_{i=1}^{j} (R_{i,j} - C^j_{\mu})^2} & j < T \\ \sqrt{\frac{1}{T} \sum_{i=1}^{T} (R_{i,j} - C^j_{\mu})^2} & j \geq T \end{cases}$$

![Fig. 2: Temporal sliding window showing reconstruction error ($R_{i,j}$) per frame ($Fr_j$) with $T = 8$](image)
$C_j^\mu$ and $C_j^\sigma$ give an anomaly score per-frame, while incorporating information from the past and future frames. A large value of $C_j^\mu$ or $C_j^\sigma$ means that the $j^{th}$ frame, when appearing at different positions in subsequent windows, is reconstructed with a high average error or with highly variable error; thus, indicating the occurrence of a fall. As this method calculates anomaly at the frame level, it is directly comparable with DAE-AN and CAE-AN. For DAE-AN and CAE-AN, the reconstruction error of an input frame is used as an anomaly score.

Window Level Anomaly: In the window level anomaly method, the score for the entire window of $T$ frames is calculated. For an input $x$ comprising of $T$ frames, this score, can be either of the following:

(i) Reconstruction error of the 3DCAE, $R(x_{i,j})$. For a particular window $i$, the mean of reconstruction error of all the $T$ frames ($W_i^\mu$) and their standard deviation ($W_i^\sigma$) are used as an anomaly score, as follows:

$$W_i^\mu = \frac{1}{T} \sum_{j=i}^{T+i-1} R_{i,j}, \quad W_i^\sigma = \sqrt{\frac{1}{T} \sum_{j=i}^{T+i-1} (R_{i,j} - W_i^\mu)^2} \quad (8)$$

(ii) Probability score of the discriminator 3DCNN, $D(x)$,

(iii) Probability score of the discriminator on the reconstructed input, $D(R(x))$ [15],

Combination of both autoencoder and discriminator scores, i.e.

(iv) $D(x) + \lambda R(x)$, and

(v) $D(R(x)) + \lambda R(x)$

The anomaly scores (iv) and (v) will have two versions each based on the mean and standard deviation of the reconstruction error, represented as $W_i^\mu - D(x) + \lambda R(x)$ and $W_i^\sigma - D(x) + \lambda R(x)$, and $W_i^\mu - D(R(x)) + \lambda R(x)$ and $W_i^\sigma - D(R(x)) + \lambda R(x)$. The − sign should not be confused with the minus sign; it only shows that this particular scores is derived from the mean or standard deviation of the reconstruction error.

The number of fall frames present in a window ($\alpha$), s.t. the ground truth label of the entire window is a fall is a hyperparameter of the method and will influence the detection of anomalies. Giving a window the ground truth as a fall with low value of $\alpha$ may result in high false alarm rate. Whereas deciding a window as a fall with high value of $\alpha$ may miss some falls. In the experiments, we varied the value of $\alpha$ from 1 to $T$ to understand the impact of choosing its appropriate value.

5 Experiments and Results

5.1 Datasets

We use the following three datasets to test the proposed spatio-temporal adversarial framework to detect unseen falls. All of these datasets contain videos captured through thermal or depth cameras. Therefore, these datasets are capable of partially or fully obfuscating the identity of the person in the video.

1. Thermal Dataset: The Thermal dataset [18] contains 9 videos with normal ADL and 35 videos containing falls and other normal activities. These videos are captured using a FLIR ONE thermal camera mounted to an Android phone with a spatial resolution of $640 \times 480$. The videos have a frame rate of either 25 fps or 15 fps, which was
obtained by observing the properties of each video. The thermal camera can protect the privacy/identity of the individual and can capture images during night conditions as well. To create sequence of windows to be given as input to the proposed spatio-temporal adversarial framework, sliding window \((T = 8)\) is performed on all video frames, resulting in 22,116 frames from 9 ADL videos. A sample of normal ADL and fall activities from the thermal dataset is shown in Figure 3.

2. **UR Dataset**: The UR dataset [2] contains 40 videos of person doing normal ADL (such as walking, sitting down, crouching down, and lying down in bed) and 30 videos with a fall in them. Two types of falls were performed by five persons from standing and sitting on the chair. These videos are captured at 30 fps using a Kinect depth sensor, which obfuscate the identity of the person. The depth map is stored in VGA resolution (640 × 480). The UR dataset has many missing pixel regions, called ‘holes’, which were filled using a method based on depth colorization [11]. The new version of this dataset obtained after filling the holes is called as UR-filled in this paper. After applying the sliding window \((T = 8)\), 8,661 windows of contiguous ADL frames were obtained for training the spatio-temporal adversarial framework. A sample of normal ADL and falls from UR and UR-filled dataset is shown in Figures 4 and 5.

3. **SDU Fall Dataset**: In the SDU Fall dataset [10], ten young men and women did six types of activities 30 times, resulting in 1800 video clips. The data shared with us contained 1197 videos, out of which 997 normal normal ADL and 200 had falls. The activities included falling, bending, squatting, sitting, lying, and walking. These videos are captured using a Kinect camera (thus hiding person’s identity) at 30fps, with video frame size of 320 × 240 and stored in AVI format. The SDU fall dataset also had holes similar to the UR dataset. However, the information on distance of depth frames is not provided with this dataset; therefore, we use an inpainting approach [1] to fill these holes, we call that data as SDU-filled. After applying the sliding window \((T = 8)\), 163, 573 windows
of contiguous frames were obtained. A sample of normal ADL and falls from SDU and SDU-filled are shown in Figures 6 and 7.

In all the datasets, there are empty frames with no person, with person entering from left, right, front far end or a full person in the scene. All the frames in all the datasets are resized to $64 \times 64$, normalized by dividing the pixel value by 255 to keep them in the range $[0, 1]$, then subtracting the per-frame means from each frame, which keeps the pixel values in the range $[-1, 1]$. The different adversarially trained methods are trained on only the normal ADL frames or their sequences. For testing, videos are presented to the trained
network containing both normal ADL and unseen fall frames (or their sequences), which were manually annotated as ground truth. Since a fall is a short event, it can only take few frames for a fall event from start to end. In our datasets, the maximum number of frames for a fall to occur was 13. Since we wanted to keep the number of frames to be a power of 2; therefore, we choose \( T = 8 \) as higher values of \( T \) would not be possible. Smaller values of \( T = 4 \) resulted in more false alarms and their results are not shown in the paper. In our implementation of the spatio-temporal adversarial learning, we use SGD optimizer with learning rate equals to 0.0002 for the 3D CNN discriminator, and adadelta optimizer for the 3D C AE. We trained our model on various values of \( \lambda \). Larger values of \( \lambda \) lead to mode collapse problem. Therefore, we choose \( \lambda = 1 \) that gave the best results. We train all the adversarial methods for a maximum of 500 epochs.

5.2 Results

Frame Level Anomaly: Table 4 shows the AUC values after applying frame level anomaly scoring method on DAE-AN, CAE-AN and the proposed spatio-temporal adversarial network (on \( C_\mu \) and \( C_\sigma \) anomaly score). The best AUC values are shown in gray color cells. We observe that the proposed method performs better than DAE-AN and CAE-AN on all the datasets, except SDU-filled with DAE-AN. The SDU dataset videos contain simple and organic activities, falls always happened from standing, besides having no furniture or background objects in the scene. We hypothesize that due to these reasons the DAE-AN and CAE-AN may be able to learn global and spatial features that may be able to detect falls comparable to the spatio-temporal network. However, the activities in the Thermal and UR datasets were complex, falls happened in various poses (e.g. falling from chair, falling from sitting and falling from standing), and the scene involved different objects in the background (e.g. bed, chair). Besides that, in the Thermal dataset, due to a person entering the scene, the pixel intensity would change values due to change in the heat in the environment. The proposed spatio-temporal adversarial learning method worked well under these diverse conditions to detect unseen falls. We also observe that all the fall detection methods perform worse on original UR and SDU datasets than their holes filled versions. This clearly shows that videos with holes are detrimental to learn normal ADL and identify unseen falls. We further observe that AUC results of the proposed approach are slightly better with \( C_\sigma \) than \( C_\mu \) for all the datasets.

| Models | Thermal | UR | UR-Filled | SDU | SDU-Filled |
|--------|---------|----|-----------|-----|------------|
| DAE-AN | 0.62    | 0.46 | 0.65      | 0.68 | **0.91**   |
| CAE-AN | 0.62    | 0.36 | 0.78      | 0.62 | 0.89       |
| \( C_\mu \) | **0.95** | 0.47 | 0.88      | 0.69 | 0.90       |
| \( C_\sigma \) | **0.95** | 0.74 | **0.91**  | 0.69 | **0.91**   |

Table 4: AUC values for different adversarial networks for each dataset (using frame based anomaly scoring).
Window Level Anomaly: Figures 8, 9 and 10 show the AUC values of the spatio-temporal adversarial learning on detecting unseen falls on Thermal, UR-Filled and SDU-Filled datasets using window level anomaly scores w.r.t. different choices of $\alpha$ from 1 to 8 (which is the maximum size of the window). The results on UR and SDU with holes were consistently worse and are not shown. We observe that for all different anomaly scores for each of the datasets, the AUC initially increases with an increase in the number of fall frames in a window (i.e. $\alpha$) and then stabilizes for higher values of $\alpha$. This is related to the fact that if a window is decided as a ‘fall’ based on very few fall frames, it would lead to many false alarms, hence lower AUC. It can be clearly seen that the anomaly score $D(R(x))$ performs worst in all the datasets. Furthermore, in UR-Filled datasets, the two other worse performing scores are $W_\sigma - D(R(x)) + R(x)$ and $D(x)$, and in SDU-Filled are $D(x)$ and $W_\sigma - D(x) + R(x)$. Other anomaly scores perform equivalent to each other. This experiment suggests that unseen falls can be detected with high AUC using window level anomaly scoring. However, the scores obtained at the discriminator or when combined with reconstruction error may not be a good candidate for detecting unseen falls.

It is to be noted that the scores of window level anomaly scoring are not directly comparable with frame level scoring method. In the frame level method, the anomaly score is calculated for every frame (occurring at different windows). Whereas in the window level method, we designate the class of the whole window instead of deciding the class of every frame across windows. Another factor in window level anomaly is the number of fall frames present in a window ($\alpha$), s.t. the ground truth label of the entire window is a fall. This parameter is varied and results are shown in Figures 8, 9 and 10. Therefore, these two types of anomaly scoring methods are not directly comparable and their separate results are discussed in the paper.

The present framework may detect other abnormal activities that significantly deviates from normal ADL as falls, such as syncope, tripping, or presence of new objects in the scene. However, on the fall datasets we tested, we get very encouraging results to support our hypothesis.

6 Conclusions and Future Work

This paper deals with identifying unseen falls in videos using a new spatio-temporal adversarial learning framework. The videos used in this paper are privacy preserving, such as thermal and depth cameras that can partially or fully obfuscate facial features of a person. This further ascertains the idea that for fall detection problem, only spatial and temporal information contained in the video is needed and not the identity revealing information (e.g. face of the person). We present a learning strategy to train this adversarial framework using spatio-temporal autoencoder and a spatio-temporal discriminator. The paper also discusses two new types of anomaly scores. The results on three public datasets suggest high performance in comparison to two other spatial adversarial methods. Encouraged by the results presented in this paper, we are currently collecting a new dataset on fall detection using multiple types of vision sensing modalities, such as thermal cameras, depth cameras, an IP camera and a RGB camera (as a baseline). These cameras will be mounted on the ceiling to represent a more realistic scenario of placing them in a home-setting. This unique dataset will be made public and will help us comparing different sensing modalities for the problem of fall detection. Furthermore, in future, we will use spatio-temporal residual networks in an adversarial setting to detect unseen falls.
Fig. 8: Variation of AUC w.r.t $\alpha$ on Thermal dataset for different models (using window based anomaly scoring).

Fig. 9: Variation of AUC w.r.t $\alpha$ on UR-Filled dataset for different models (using window based anomaly scoring).

Fig. 10: Variation of AUC w.r.t $\alpha$ on SDU-Filled dataset for different models (using window based anomaly scoring).
References

1. Bertalmio, M., Bertozzi, A.L., Sapiro, G.: Navier-stokes, fluid dynamics, and image and video inpainting. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, vol. 1, pp. I–355–I–362 vol.1 (2001). DOI 10.1109/CVPR.2001.990497

2. Bogdan Kwok, M.K.: Human fall detection on embedded platform using depth maps and wireless accelerometer. Computer Methods and Programs in Biomedicine 117, 489–501 (2014)

3. Eide, A.W.W.: Applying generative adversarial networks for anomaly detection in hyperspectral remote sensing imagery. Master's thesis, NTNU (2018)

4. Igual, R., Medrano, C., Plaza, I.: Challenges, issues and trends in fall detection systems. Biomedical engineering online 12(1), 66 (2013)

5. Ji, S., Xu, W., Yang, M., Yu, K.: 3d convolutional neural networks for human action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(1), 221–231 (2013)

6. Khan, S.S., Hoey, J.: Review of fall detection techniques: A data availability perspective. Medical engineering & physics 39, 12–22 (2017)

7. Khan, S.S., Madd, M.G.: One-class classification: taxonomy of study and review of techniques. The Knowledge Engineering Review 29(3), 345–374 (2014)

8. Lawson, W., Bekele, E., Sullivan, K.: Finding anomalies with generative adversarial networks for a patrolbot. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 12–13 (2017)

9. Lee, S., Kim, H.G., Ro, Y.M.: Stan: Spatio-temporal adversarial networks for abnormal event detection. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1323–1327. IEEE (2018)

10. Ma, X., Wang, H., Xue, B., Zhou, M., Ji, B., Li, Y.: Depth-based human fall detection via shape features and improved extreme learning machine. IEEE Journal of Biomedical and Health Informatics 18(6), 1915–1922 (2014). DOI 10.1109/JBHI.2014.2304357

11. Nathan Silberman Derek Hoem, P.K., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: ECCV (2012)

12. Nogas, J., Khan, S.S., Mihailidis, A.: Fall detection from thermal camera using convolutional lstm autoencoder. In: Proceedings of the 2nd workshop on Aging, Rehabilitation and Independent Assisted Living, ICAI Workshop (2018)

13. Nogas, J., Mihailidis, A., Khan, S.S.: Deepfall-non-invasive fall detection with deep spatio-temporal convolutional autoencoders. arXiv preprint arXiv:1809.00977 (2018)

14. Ravanchkhosh, M., Nabi, M., Sanguineti, E., Marcenaro, L., Regazzoni, C., Sebe, N.: Abnormal event detection in videos using generative adversarial nets. In: 2017 IEEE International Conference on Image Processing (ICIP), pp. 1577–1581 (2017)

15. Sabokrou, M., Khaledi, M., Fathi, M., Adeli, E.: Adversarially learned one-class classifier for novelty detection. In: In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3379–3388 (2018)

16. Schlegl, T., Seeböck, P., Waldis, S.M., Schmidt-Erfurth, U., Langs, G.: Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: International Conference on Information Processing in Medical Imaging, pp. 146–157. Springer (2017)

17. SMARTSRISK: The economic burden of injury in canada. http://www.parachutecanada.org/downloads/research/reports/EBI2009-Eng-ExecSumm.pdf/ (2009). Accessed on 18thth August, 2018

18. Vadivelu, S., Ganesan, S., Murthy, O.R., Dhall, A.: Thermal imaging based elderly fall detection. In: ACCV Workshop, pp. 541–553. Springer (2016)

19. Yarlagadda, S.K., Guerra, D., Bestagini, P., Maggie Zhu, F., Tubaro, S., Delp, E.J.: Satellite image forger detection and localization using gan and one-class classifier. Electronic Imaging 2018(7), 1–9 (2018)

20. Yu, Y., Qu, W.Y., Li, N., Guo, Z.: Open-category classification by adversarial sample generation. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence, pp. 3357–3363. AAAI Press (2017)