IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 19 August 2023, accepted 25 September 2023, date of publication 2 October 2023, date of current version 10 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3321192

RESEARCH ARTICLE

Visual Fall Detection From Activities of Daily Living for Assistive Living

SAMYAN QAYYUM WAHLA AND MUHAMMAD USMAN GHANI
Department of Computer Science, University of Engineering and Technology Lahore, Lahore 54890, Pakistan

Corresponding author: Samyan Qayyum Wahla (samyan.qayyum@uet.edu.pk)

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Advanced Studies and Research Board, Office of Research, Innovation and Commercialization, University of Engineering and Technology Lahore, Pakistan.

ABSTRACT Health facilities have increased life expectancy, a key factor for the growth of the elderly population. Elderly people are at increased risk of falls, causing physical and psychological damage. Falls occur rarely compared to other activities of daily living. Due to such a class imbalance, supervised techniques are not the solution for fall detection systems. In addition, domain-level features for the fall activity are hard to generalize due to their diversity. In this work, the fall detection problem is formulated as anomaly detection in the time series where deviation from the activities of daily living is computed. On the basis of the deviation score, a fall is detected. We propose TCHA, Temporal Convolutional Hourglass Autoencoder, to learn spatial and temporal features from the videos. Hourglass units in the Temporal Convolutional Encoder help us extract multiscale features by expanding the receptive fields of neurons, reducing the information loss in deep learning methods. The proposed methodology is evaluated on the five data sets, including a compiled data set from publicly available Toyota Smarthome data set and four benchmarked datasets that include the UR-Fall dataset, IMVIA dataset, SDU dataset, and Thermal Fall dataset. Our methodology shows 4.1% superior results to existing state-of-the-art methods for unseen falls.

INDEX TERMS Autoencoders, computer vision, fall detection, assistive living, visual anomaly, unsupervised, time series data.

I. INTRODUCTION

In the modern era, the quest for a better lifestyle has increased, helping the human being in the activities of daily living. On the other hand, life expectancy has also increased. As a result, the population of the elderly is growing. According to the report of the World Health Organization (WHO) on the World Aging Population, approximately one-fourth of the population will be over 60 years old by the end of 2050 [1]. A large portion of the population older than 60 years demands more assistive resources to be looked after and taken care of. Based on these statistics, researchers are actively working on domains that address the elderly population directly.

Researchers are developing systems that autonomously monitor elderly people in their daily routine tasks and help them without the assistance of the caregiver. The most important cause of concern for older people and patients with chronic diseases is the fall activity from activities of daily living(ADL). According to the World Health Organization(WHO), falling is the leading cause of unintentional injuries and deaths. Individuals older than 60 years experience the highest number of fatal falls, which varies based on country, age, and disease. More than 80% of fall-related deaths occur in low and middle-income countries. People in under-developed countries experience more incidents of falls compared to developed countries. Diseases that can increase the risk of falls are stroke, heart disease, loss of balance, and Alzheimer's disease.

Falling is considered an important health problem with age and requires immediate medical attention. Usually, the patient keeps lying on the floor, which can cause additional problems in the event of an unattended fall. Automatic fall detection systems can reduce such consequences by broadcasting the

The associate editor coordinating the review of this manuscript and approving it for publication was Prakasam Periasamy.

information to the concerned ones. A fall detection system should generate alerts for individuals who experience fall activity and face permanent injury or even death.

A plethora of research has been done on fall detection systems, but the problem still needs a lot of attention based on its importance and sensitivity. At a higher level, the research is divided into three modalities; vision-based devices, wearable sensors, and ambient sensors. This research is focused on the vision modality because vision-based methods are the most accurate, usually for activity detection and specifically for fall detection systems. It has been proven that ambient or wearable sensors do not add more information than vision-based devices [2]. In addition, it is inconvenient for the elderly to use wearable sensors.

The research in fall detection systems is further divided into sub-modalities for vision-based devices. Modalities include RGB cameras, thermal cameras, and depth cameras. Each vision-based modality has its own pros and cons. RGB modality helps to extract color and texture-based information in addition to global features. In comparison, depth and thermal cameras protect a person's privacy, but some of the important information related to background scene and context is lost. The algorithms in this research can be adapted to any of the visual modalities. However, we prefer the RGB modality while integrating the architecture to protect the privacy of elders and chronic diseased patients.

Each activity/event is a combination of unit actions or composite actions. It is really hard to provide a universal definition of an unexpected fall. The best way to approach a problem where features are explicitly unknown is to use deep learning approaches with implicit feature extraction. The data is presented to the deep learning algorithms to learn the features from provided examples with an approximately balanced number of examples of each event. The issue with the fall detection data is that the fall activity is unexpected and unintentional; it is almost impossible to balance the examples of fall detection with the normal activities of daily living. Each individual in the aging community produces approximately 32 million activities per year, of which only 2.6 falls are produced in the data [3]. Due to such high skewness, the problem of fall detection cannot be approached with algorithms that require balanced data.

With the evolution of algorithms in machine learning and deep learning, different algorithms are used to detect the fall activity of individuals accurately. In the bird's eye view, the algorithms are divided into two main categories; fall detection as a supervised problem and fall detection as an unsupervised problem. Supervised algorithms use data augmentation, simulation, and other techniques to reduce class imbalance. This type of category is divided further according to the characterization of the input. The work of some researchers is based on hand-crafted features, while others use deep learning to learn the features. Furthermore, most available datasets are simulated and contain the generated rule-based data. The nature of falls cannot be predicted, and all scenarios cannot be handled in the provided data set. There are very few datasets that contain some real falls.

We propose an unsupervised technique with the use of autoencoders while considering the fall activity as an anomaly. The encoder mainly classifies the activity, and the decoder discriminates the reconstructed image with the ground truth input. The high reconstruction error infers a high probability of anomaly and a high probability of falling. In this research, the real fall data is used. The problem is approached as an imbalanced data problem, where the algorithms are trained on the normal activities of daily living, and fall activity is taken as an anomaly. Our contributions to this research are three fold.

- We provide framework to detect anomalies from assistive living data in general and specifically fall activity where data of anomalous activity is not available.
- Our proposed method shows an improved detection rate for unseen falls as compared to the available state-of-the-art methods.
- We train the system on more than 50 activities of daily living, which solves the problem of false alarms, whereas existing algorithms use limited activities of daily living as Normal class, which cannot generalize the normal behaviour of the individual under observation.

This research article is organized as follows; Section II discusses related work. Section III contains the datasets used for this research work. Section IV explains the state-of-the-art methodology, and Section V contains the results and comparisons with the existing algorithms and benchmark datasets. Finally, the conclusion and further improvements are discussed in Section VI.

II. RELATED WORK

The proposed research considers fall detection as a case study to validate anomaly detection in the videos of assistive living. In this section, we discuss and compare existing fall detection algorithms.

The most common category of algorithms for fall detection systems is based on Long-Short-Term Memory (LSTM), Recurrent Neural Network (RNN), and Gated Recurrent Units(GRU). Features are extracted explicitly in these algorithms, such as spatial features, shape deformation, and motion features. The features are extracted implicitly using the 2D Convolution Neural Network(CNN) in other works.

Ziwei et al. propose a video-based fall detection system based on CNN. Using CNN, they identified the computational cost of the big videos in previous algorithms. Additionally, complex lighting conditions and backgrounds interfere with the extraction of features. Researchers estimated human pose from the video from where the 2D poses are extracted and lifted to 3D poses, due to which computation cost is decreased. In addition, dilated convolutions increase the receptive field of the neural network. The algorithm is evaluated on the NTU dataset and shows superior results to existing algorithms in this category [4]. In [5], a similar approach for the classification of fall activity is used where the CNN-based human pose estimator is used in combination with the 3D ground estimation. The key novelty of this research is that the reasoning module formulates several measures to infer whether a person has fallen or not. Reference [6] use the Gated Recurrent Neural Network to obtain the temporal dependency of human motion after extracting the video frames using 2DCNN. The binary class entropy function and the sigmoid classifier tune the network and predict the results, respectively. A pre-trained version of VGG-16 and VGG-19 is tuned for the computation model. The results are evaluated on the benchmark data set, the Multiple Camera data set and the UR fall detection data sets to obtain the precision of 95% fall detection.

Fall detection is approached as a supervised problem with input characterization differently, where the fall detection algorithm is based on the symmetry principle [7]. OpenPose algorithm [8] extracts joint points of humans such as the speed of descent at the center of the hip joint, the angle of the center line of the human body with the ground, and the width-to-height ratio of the rectangular external body.

Guan et al. use the LSTM to characterize input differently and detect human features using a pose estimator to make the algorithm lightweight. Results are evaluated on the Multiple Cameras dataset and UR-Fall dataset. The results outperform the OpenPose algorithm [9].

Musci et al. studied online falls using the Recurrent Neural Network and Wearable Sensors. The work divides the events into three classes; Fall is the state when the person is experiencing the transition to fall, and ALERT state defines the interval in which a person is in a dangerous state. BKG class indicates that the person being monitored is in a controlled state. BKG handles all the activities of daily living. SisFall is used for the training and testing process using deep learning techniques and RNN [10].

Similarly, there are a lot of other works that use supervised Deep Learning techniques to identify falls. The fall events to be recognized are defined prior to the system deployment. Santos et al. did similar experiments on accelerometers for human fall detection. Data Augmentation is used to achieve the best results [11].

One of the CNN categories approaches uses transfer learning for fall detection. The work is on still images; data of fall detection is balanced using data augmentation and then passed to a pre-trained CNN model. The model is tested on the depth images with a recall score 99% [12].

In the other category, hand-crafted temporal or spatial features are provided as input to deep learning methods. These hand-crafted features also contribute to the solution of fall detection to a large extent.

In [13], researchers propose an algorithm based on the motion image of the human. Two categories are defined for the classification problem; Abnormal Activity, such as falling, and normal activities, such as routine life activities. Algorithms track the motion of a person for a moment to create a Motion History Image(MHI). Shape deformation is detected using MHI and speed of shape change in the history image. A large amount of deformation infers abnormal activity. The fall detection Dataset by Antoine Trapet is used. Further, the fall detection is compared with evaluation results of overlapping activities such as sitting on the chair, bending the shoe rack, and lying down. The proposed system efficiently differentiates normal and abnormal activities.

Chhetri et al. propose a system to improve fall detection accuracy in dynamic lighting conditions. Enhanced dynamic optical flow is integrated to achieve the desired results and to rank optical flow data by max pooling. Classification accuracy improves by 3% and time by 40ms to 50ms. Further, dynamic optical flow summarizes the whole video of human activity into a single image, which ultimately can reduce the computation cost [14].

In [15], an algorithm is proposed to detect falls based on geometric features and CNN. The head is given importance for the geometric features. Two elliptical contours for the head and torso are used to calculate three geometric features for each ellipse. These features include long and short axis ratio, vertical velocity, and orientation angle. Geometric features are then passed to shallow CNN to reduce the computation cost. The self-generated dataset is used to evaluate the results. Researchers claim to differentiate the overlapping activities more efficiently.

Kavya et al. present real-time algorithm to efficiently detect the fall using ground point and person body rectangle. Ground point is estimated in the RGB images using the Gabor filter on texture segmentation, and the person is detected using the Kalman filter. The angle between the person and the ground's estimated point helps detect the fall. The results are evaluated on two publicly available datasets; the UR-Fall and Fall detection datasets (FDD). 90% accuracy is reported in the research work [16].

In [17], an algorithm based on a Multivariate exponentially weighted moving average (MEWMA) monitoring scheme is presented. Person identification is performed based on segmentation using a background template. Five virtual lines on the silhouette are drawn to calculate the features. Then, MEWMA is applied to detect the small shifts. One issue that the MEWMA poses is that it cannot differentiate the overlapping activities similar to falls, which is then resolved using the SVM classifier as a final stage. Results are evaluated on the UR-Fall Detection data set and Fall Detection dataset.

One of the fall detection systems categories uses the multimodal approach, i.e., multiple input modalities, to reduce false positives and enhance the system's accuracy.

Yves et al. [18] propose the different topologies of multimodal convolutional neural networks. Different channels of CNN are used for each modality, such as vision and sensors. ID kernels are used for temporal windows, while 2D kernels with 2D convolutions are used for spatial feature extraction. UP Fall detection and UR Fall detection dataset [19] are used, which contain the synchronized input of accelerometers and RGB cameras. All the categories of fall detection in the dataset are marked as positive samples. Other activities of daily living are labeled as negative samples to approach the problem as a binary classification problem. Recent fall detection advancements use auto-encoders, which are most relevant to our work.

Metha et al. approach fall detection using unsupervised algorithms due to the rarity of falls, as in supervised algorithms, there is a large class imbalance. Thermal sensing modality is used to acquire the input images. The computational model is based on two-channel 3D auto-encoders; the first is to reconstruct thermal images, and the second is to reconstruct the optical flow images. In this way, both channels deal with spatial and temporal features at a time, respectively. Another contribution is to track the region of interest for fall detection to reduce the computation cost. A publicly available dataset of thermal images is used to evaluate the results. High reconstruction errors mean a high probability of fall detection. Results are superior to existing baseline algorithms [20].

Iguchi et al. propose that a large amount of fall detection data is tiresome to collect, due to which supervised methods cannot perform better for fall detection. A convolutional auto-encoder unsupervised algorithm is proposed using an IMU sensor. Monochrome images are generated for motion data in the training process and compared to fall data with the configuration of threshold value on the monochrome images. Two datasets are considered to evaluate the computational model, MobiAct, and the other one self-generated data [21].

In [22], Nogas et al. proposes considering the fall detection problem as an anomaly detection problem. Data on fall detection is rare, and it is very difficult to extract the domain-specific features to identify falls. Researchers proposed the temporal convolutional auto-encoder model to learn the Spatio-temporal model. Data from normal activities of daily living is passed as input to the model. Fall activity is considered an abnormal activity of daily living. An anomaly scoring method is defined to mark the activity as an anomaly that combines the reconstruction score of frames across a predefined temporal window. 3D convolutions are used for the encoding and decoding process. Results are evaluated on Thermal videos and RGB datasets, which are comparable and superior to previous algorithms.

Cai et al. also use auto-encoders but do not approach the problem as unsupervised. Instead, the encoder part of the algorithm performs fall detection as a primary task. The representativeness of the features is enhanced using the Hourglass Convolutional Neural Network as a secondary task. Hourglass units obtain the multi-scale features to preserve the spatial information at each resolution. Three branches, the hourglass identity branch, the mapping branch, and the residual mapping branch, are used to obtain images at original, low, and high, respectively. These three branches are then integrated to obtain the posture and motion of the human body [23].

As discussed, much work has been done for the detection of falls. However, there are still open challenges that need to be handled to make fall detection more accurate and work in the production environment.

1) Table 1 shows that a fraction of research works approach the fall detection problem as an unsupervised problem. There is a high probability that the nature of the fall is unseen or seen with rare data in the production environment, due to which fall detection cannot perform better as a supervised problem.
2) Most of the work on unseen falls is done using wearable sensors, which are not convenient for older people. Moreover, wearable sensors for activity detection are considered dead-end due to much information abstraction. These sensors cannot add more information to the characterization of falls [2].
3) The work on the unseen falls use the limited data for normal activities of daily living, which in turn can generate false alarms in case of deviation from the normal activities.
4) Most of the work on vision-based modalities is invasive, that is, they use the RGB cameras for the feature extraction without providing the mechanism to protect the privacy of the person being monitored. There are some non-invasive methods as well, but they lose the information of color and texture-based features.

We propose an unsupervised computational model that is based on vision modalities. Vision modalities can capture much information to characterize a fall. Moreover, the model is also capable of detecting unseen falls. Our algorithm benefits from the RGB cameras as well as the ability to be non-invasive. The proposed architecture solves the problem of ensuring that the privacy of the person under observation is protected whenever a video stream is transferred to the network.

III. DATASET

The nature of the datasets used in the proposed methodology discussed in the next section contains the videos of the cameras mounted parallel to the subject being monitored in the indoor environment. The proposed environment for the detection of falls is shown in Fig. 1.

Four publicly available datasets and one custom dataset compiled from public dataset are used to validate the proposed study. The best benchmarked available dataset is UR-Fall Detection, which contains 30 fall videos captured from two cameras; ceiling-based (cam0) and parallel to the floor(cam1). Videos from the relevant camera are sampled. Further, videos for the normal activities of daily living are also sampled from the same dataset, which contains 40 videos of normal activities. UR-Fall dataset is available publicly [33].

The SDU dataset consists of 1800 depth videos recorded by 10 individuals, including both men and women. Videos were

TABLE 1. Comparison of state of art fall detection techniques.

Research	Modality	AC	USP	TI	PI	Dataset	Evaluation	Comments
V Garrapally et al. 2023[24]	Video	CNN and LSTM	✗	Temporal Autoencoder	✗	UP-Fall dataset	Accuracy 98.59%	W1, W2, W3,W5
S Mobsite et al. 2023[25]	Video	ConvLSTM and Xception network	✗	Temporal Autoencoder	✗	UP and UR Fall dataset	F1-score of 97.68% and 97.85%	W1, W2, P4,W5
A Khatun et al. 2023[26]	Accelerometer	SVM and KNN	✗	Temporal Autoencoder	✗	Smartphone accelerometer dataset	Accuracy 99.54%	W1, W2, W4,W5
KC Tran et al. 2022[27]	Video (RGB)	SVM	✗*	Temporal Autoencoder	✗	self-collected dataset	Accuracy 90%	W1, W2,W3,W5
M Villar et al. 2022[28]	tri-axial Accelerometer	LSTM	✗	Temporal Autoencoder	✗	UCI-FALL dataset	Accuracy 0.9968 with data augmentation	W1, W2, W4,W5
DR Beddiar et al. 2022[29]	Video(RGB)	SVM, TCN, and LSTM	✗	Temporal Autoencoder	✗	Le2i and the UR FD dataset	Precision, Recall and F_score	W1, W3
Guan et al. 2021[9]	Video (RGB)	LSTM	✗	Time Series Joint-Point Features	✗	UR-Fall Dataset	mAP 73.3	W1, W3, P4,W5
Ziwei Chen et al. 2021[4]	Video (RGBD)	CNN-Fall Detection Network	✗	1-D dilated temporal convolution	✗	NTU RGB Dataset	Accuracy 99%	W1, W2, W3,W5
Galvao et al. 2021[18]	Video (RGB), Accelerometer	CNN- One channel for one modality	✗	LSTM	✓	UR Fall detection ,UR-Fall	Accuracy 99%	W1, W3
Sultana et al. 2021[6]	Video (RGB)	LSTM	✗	GRU	✓	UR FDD,Multi-Cam-FD	Accuracy 99%	W1, W2, W3,W5
Chhetri et al. 2021[14]	Video (RGB)	CNN	✗	Optical Flow for variable lighting conditions	✗	URFD, Multi-Cam, FDD	Accuracy 99%	W1, W2, W3
Metha et al. 2020[20]	Video (Thermal)	Auto Encoders	✓	Optical Flow	✗	Thermal Images Data Set	AUC, Precision, Recall	P1, W2, W5
Thummala et al. 2020[30]	Video (RGB)	Machine Learning	✗	Motion History Image	✗	Fall detection Dataset by Antoine Trapet	Accuracy 95%	W1,W2,W5
Nogas et al. 2020	Video	Auto Encoders	✓	Temporal Autoencoder	✗	Thermal, UR-Fall, SDU	Anomaly Score	P1,P2, W5
Chenguang Yao et al. 2020[15]	RGB	CNN- Head Torso Ellipse	✗	Time Series Motion Features	✗	self-collected (30K frames)	Fall Detection Rate 90.5%	W1, W2,W5
Kayva et al. 2020[16]	Video (RGB)	Machine Learning	✗	Rate of change of skeleton angle	✗	URFD, FDD	Accuracy 90.5%	W1, W2,W5
Xi Cai et al. 2020[23]	Video (RGB)	Auto Encoders	✗	LSTM	✓	UR-Fall Dataset	Accuracy 96.2%	W1, W3,W5
Weiming ChenS et al. 2020[31]	Video (RGB)	Machine Learning	✗	Speed of change of each action	✗	self-collected dataset	Fall Detection Rate 97%	W1, W2,W5
Musci et al. 2018[10]	Wearable Sensors	LSTM	✗	RNN	✗	SisFall	Precision	W1, W2, W4
Haraldsson et al. 2018[32]	Video(RGB)	Machine Learning	✗	Motion History Image	✗	FDD	Accuracy 92%	W1, W2, W3,W5
Ours-Current	Video (RGB)	Auto Encoders	✓	Temporal Autoencoder	✓	SDU,UR-Fall, IMVIA, Thermal,Custom large dataset	AUC, Specificity, Sensitivity	P1, P2, P3, P4

PI: Preserve Information in DNN, **USP**: Does the nature of problem is Unsupervised?, **TI**: Temporal Information. **AC**:Algorithm Category
W1: Methodology cannot work on unseen falls, **W2**: Methodology is evaluated on simulated and augmented data, **W3**: Overfitting due to training on the background scene as well, **W4**: Use of accelerometer sensors cannot provide rich information of activities, **W5**: Cannot handle the problem of false alarms
P1: Method is capable to handle unseen falls , **P2**: Method is evaluated on real falls, **P3**: Large number of ADL classes as Normal class reduces false alarms, **P4**: Deep Learning/ Machine Learning model is trained after segmentation of the human to reduce overfitting

recorded at a frame rate of 30 frames per second and have a resolution of 320 × 240 pixels. They are stored in AVI format. Data were captured using a Microsoft Kinect camera [34]. The dataset includes various activities such as sitting down, lying down, bending, squatting, and falling. Since falls are rare, individuals intentionally fall to capture these instances.

FIGURE 1. Proposed indoor environment settings for fall detection monitoring.

Each action was performed 30 times by each individual, resulting in multiple instances of each action. These actions were performed under different conditions, such as holding an object in different directions.

The IMVIA dataset is created using a video surveillance camera that is typically used in environments like elderly homes and office rooms. The videos are recorded with 25 frames per second and have a resolution of 320 × 240 pixels. The dataset contains a total of 191 videos that include challenges we encountered in real-time environments, such as illumination, occlusion, and textured background [35].

The thermal dataset consists of a total of 44 videos, with 9 videos capturing normal activities of daily living (ADL) and the remaining 35 videos featuring falls [36]. Videos were captured using a FLIR ONE thermal camera, and the recordings were recorded in a room with a single view, with a resolution of 640 × 480 pixels. Falls occur in different positions and are recorded in the video dataset, such as falls during standing positions or during any normal activity.

The custom data set contains normal activities from the Toyota SmartHome(TSH) data set and fall sequences from the UR-Fall dataset. Toyota's SmartHome data set is the largest data set for activities of daily living for the elderly [37]. It contains more than 50 classes of activities for 18 individuals.

Two classes are defined to categorize all the videos for each dataset.

- **NORMAL Class** contains all the activities which are considered to be out of danger, such as drinking water and taking medicine, etc.
- **ABNORMAL Class** consists of the images in which the subject being monitored needs immediate assistance, such as fall activity.

Figure 2 contains the extracted frames from the sampled ADL videos of the dataset used, and Figure 3 contains the sample frames from fall videos. Processed datasets during the current study are available from the corresponding author on reasonable request.

IV. METHODOLOGY

The solution to the fall activity is approached as an unsupervised problem due to the rarity of the occurrence of these activities in the activities of daily living and a large class imbalance. The biased distribution of the abnormal class and the normal class causes over-fitting in deep learning methods. Secondly, the proposed research guarantees to protect the privacy of the person while taking advantage of the RGB and texture-based features at the same time. The benefits of using an RGB camera and de-identifying the person in the videos are achieved through the architecture shown in Figure 4, where texture-based features are extracted on the processing node placed in the environment at the consumer end, and images are de-identified through segmentation of the persons and transmitted to the central processing unit.

The abstract view of the modeling approach is shown in Figure 5. Input from a video stream is converted to a sequence of video frames and passed to the pre-processing module for segmentation and noise Convolutional Hourglass Autoencoder(TCHAE), which is trained on normal activities of daily living and tested on fall videos. A higher reconstruction error indicates the detection of a fall event.

A. INPUT ACQUISITION

The proposed system requires the RGB video stream as the input of the camera mounted parallel to the subject being observed. The decision for the usage of the RGB camera instead of other video modalities is made due to three factors; some video modalities such as the thermal camera detect heat and can best work in a moderate environment. Some countries experience extreme weather, where heaters are installed in rooms, and in the hot environment, temperature in the daytime can affect thermal imaging. Secondly, the use of thermal cameras limits the features that can be extracted. Instead, we use RGB cameras and solve privacy issues using distributed computing between the central processing server and the processing node installed in the environment. Thirdly, thermal sensors are more expensive than RGB cameras.

B. PRE-PROCESSING

Frames are extracted from the video stream, and basic pre-processing operations are applied, including normalization, segmentation, and de-identification.

1) SEGMENTATION

The success of computer vision tasks, especially human activity detection, depends on segmentation approaches. In this work, we adopt a state-of-the-art human segmentation approach in which the segmentation of the person on the input frame is performed using the point-based rendering technique proposed by Alexander Kirillov [38]. The central idea of this work is to view image segmentation as a rendering problem and to adapt classical ideas from computer

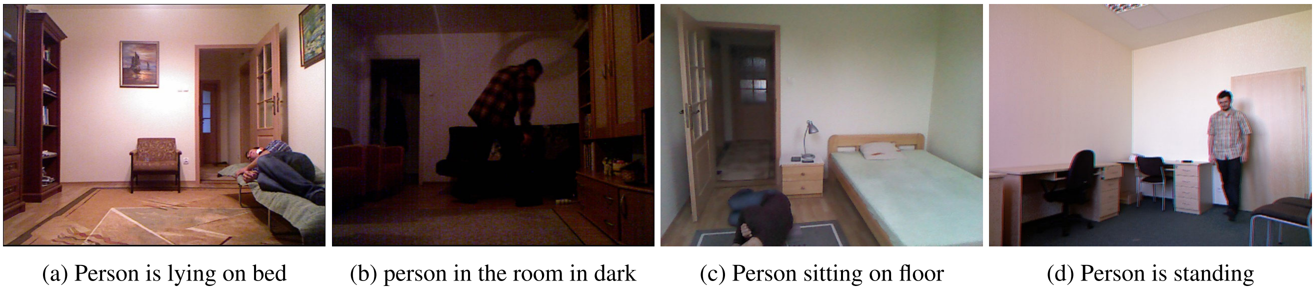
(a) Person is lying on bed (b) person in the room in dark (c) Person sitting on floor (d) Person is standing

FIGURE 2. Highlighted normal class(ADL) frames from the compiled dataset of fall detection.

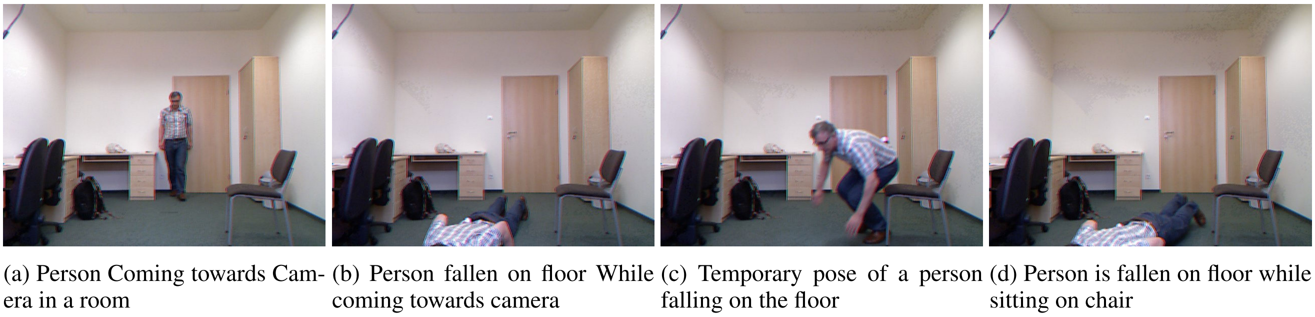
(a) Person Coming towards Camera in a room (b) Person fallen on floor While coming towards camera (c) Temporary pose of a person falling on the floor (d) Person is fallen on floor while sitting on chair

FIGURE 3. Abnormal class(Fall) frames from the compiled dataset of fall detection.

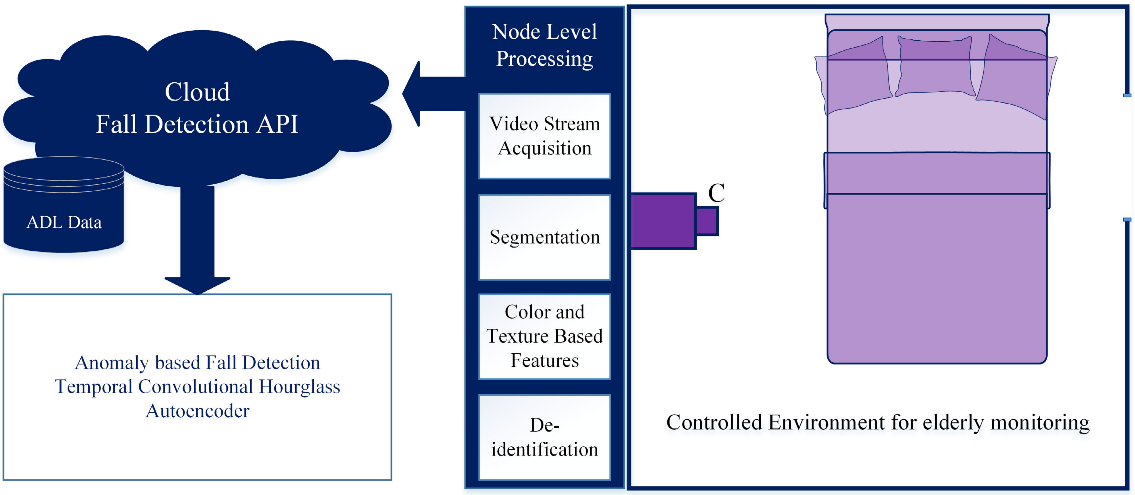

FIGURE 4. Architecture to protect the privacy of an individual in vision-based methods.

graphics to efficiently render high-quality label maps. It uses a subdivision strategy to adaptively select a non-uniform set of points at which to compute labels. We adopt an instance-based segmentation technique on top of the point render approach. In this way, the crisp masks of a person being monitored can be highlighted. Segmentation for a person in the UR-Fall dataset is shown in Figure 6b.

2) DE-IDENTIFICATION

After the image is segmented, the stream will be sent to the cloud to apply the auto-encoder for the detection of abnormal classes. Once the segmentation is completed on the node camera, we make sure to de-identify the image by applying a white mask on a black image, as shown in Figure 6c. As seen in de-identified images, there is some noise as well in terms of small contours. The small contours are removed and only the largest contour is retained in the output image using equation (1).

$$C = \arg\max_{j}(C_j) \tag{1}$$

where C_j is the enumeration over the segmented contours, and C is the largest contour to be retained, as shown in Figure 6d.

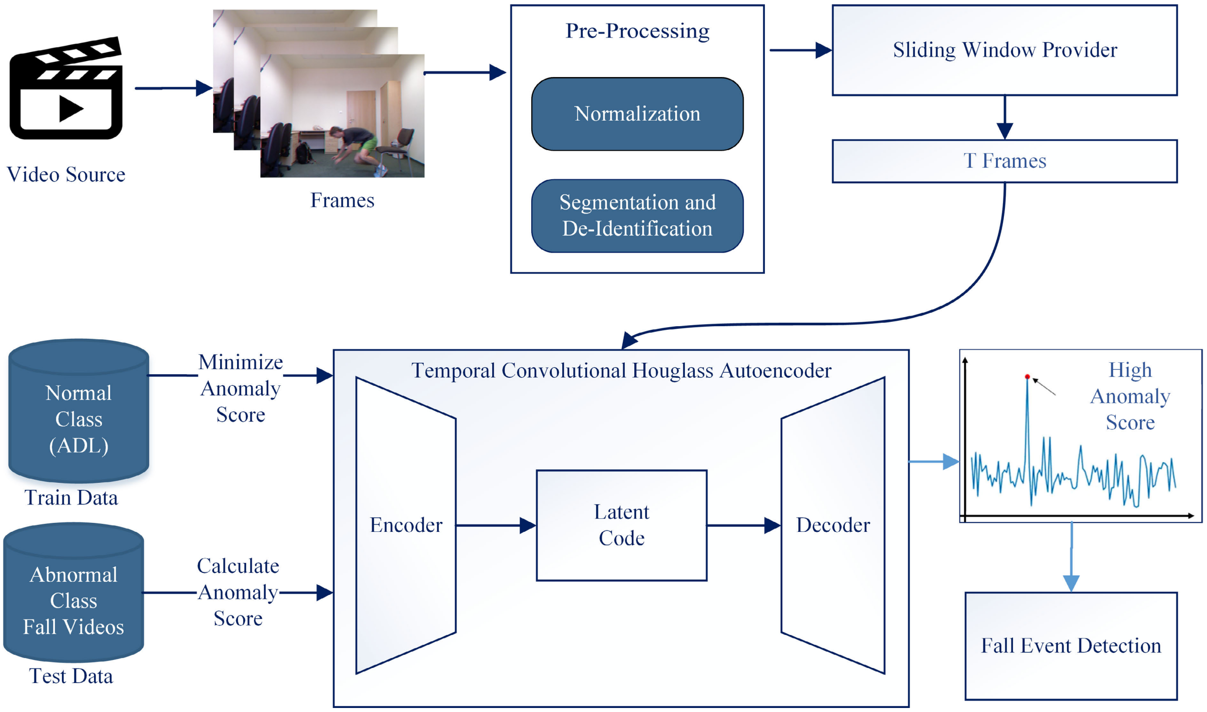

FIGURE 5. Methodology for fall detection.

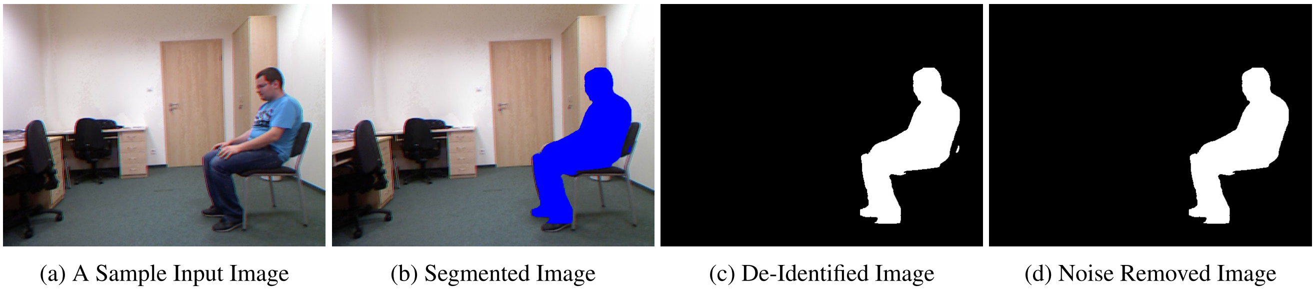

FIGURE 6. Preprocessing intermediate stages on the UR-fall dataset.

C. TEMPORAL SLIDING WINDOW

Extracted frames from the video are used to sample the frames W for processing by the computational model. These W frames are sampled by applying a temporal window to the sequence of video frames. The number of windows L generated per video is the key component in tuning the accuracy and computation power required by the system. The total number of temporal windows L can be calculated using equation 2.

$$L = \left\lceil \frac{T - (W * U)}{S} \right\rceil + U \tag{2}$$

where T is the total number of extracted frames from a video, W is the number of frames covered in one window, and S is the shift of the window from one frame to another frame. U is the frame skip within the one temporal window. Examples of the temporal sliding window are shown in Figure 7.

The smaller the size of the temporal window W, the higher the samples are generated to be processed by the computation model, and the computation model cannot increase the receptive field to include a larger number of frames to have a gist of a single activity. The size of the window W is the most discussed issue when extracting the motion characteristics on the sequence of frames for a video. Many experiments have been performed to define the universally recommended value of W. Schindler recommends $W = 7$ with evidence that motion features can be extracted more accurately on this value for atomic actions. Atomic actions are movements that cannot be subdivided further and can be completed in seven frames, which usually constitute one-third part of a second [39]. Fall is not an atomic value, but it is a combination of different atomic actions based on the scenario for which more than seven frames are required. We propose $W = 8$ with the skip frames $U = 1$ and $S = 1$ to cover the view of the complete fall activity. The reason for $W = 8$ is to

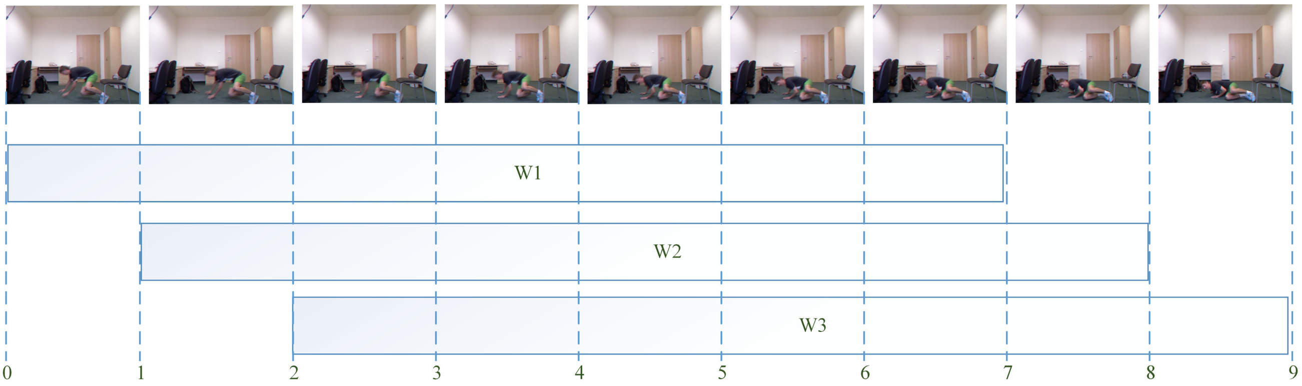

FIGURE 7. Temporal window over a video sequence with $S = 1$, $U = 1$ and $W = 8$.

make the computations more convenient at the power of 2. Values other than $W = 8$ cannot cover the complete action of the individual and hence affect the detection rate of activity detection and falls in our case. We have experimented with multiple values of the W, U and S, the results are referred to in Table 4.

D. TEMPORAL CONVOLUTIONAL HOURGLASS AUTOENCODER (TCHAE)

Temporal Convolutional Hourglass Autoencoder(TCHAE) consists of three parts; encoder, decoder, and Latent Code. In this work, the convolutional autoencoder is trained to reconstruct the frame sequence. The encoder receives a continuous stream of images that is passed through it and sent to a decoder which takes the compressed latent code and reconstructs the original sequence. The encoder is used to accept a single element of the input sequence at each time step, then it processes it, collects information for that element, and propagates it to the next part. The intermediate vector is the final internal state produced from the encoder part of the model. It contains information about the entire input sequence to help the decoder make accurate predictions. Given the entire sequence, the decoder predicts an output at each time step. TCHAE helps in capturing features from a given sequence of images and identifying the difference between input and constructed output. Furthermore, the proposed model can extract multiscale features when combined with the residual units. In our work, fully connected residual layers are used, which results in a large connection of parameters used to learn spatially global features and temporal features as well.

1) ENCODER

The encoder in the proposed model is used to find the smallest possible representation of the data that it can store, extracting the most prominent features of the original input frame sequence and representing it in a way the decoder can understand. The encoder takes the input data, that is, a sequence of images, and uses them to generate an encoded version of the input in the form of compressed data. We use this compressed data to send it to the next part of the model, where it is decoded and reconstructed. The input is encoded by a sequence of 3D residual layers and 3D pooling layers. 3D convolutions operate with stride of $1 \times 1 \times 1$ and no padding. The max-pooling layers use padding, with stride and kernel dimensions $2 \times 2 \times 2$. This means that each dimension (temporal depth, height, and width) is reduced by a factor of 2 with every max-pooling layer. This process is repeated for three levels of depth. The activation function f is set to ReLU for hidden layers in both encoding and decoding.

2) HOURGLASS RESIDUAL UNIT

A residual unit is a stack of layers that are set in such a way that the output of a layer is taken and added to another layer in the block. After adding it together with the output of the corresponding layer in the main path, non-linearity is applied to it. This bypass connection is known as the shortcut or the skip connection in a residual layer and plays an important part in increasing the accuracy. In our methodology, hourglass residual blocks are introduced to perform the model efficiently on the multi-scale input. To handle all the resolutions, we introduce three branches in the Hourglass Residual Unit; Identity resolution branch r_1 is used to retain information in the original resolution, high-resolution branch r_2 is used to extract features from the images greater than the original resolution, and low-resolution branch is used to extract features from the images having a resolution lower than the original resolution. The final output r of the residual block can be represented as the equation(3).

$$\begin{aligned} r &= r_1 + r_2 + r_3 \\ &= \sigma(W_1 * x) + u(\sigma(W_3 * \sigma(W_2 * p(x)))) \\ &\quad + \sigma(W_6 * \sigma(W_5 * \sigma(W_4 * x))) \end{aligned} \tag{3}$$

where x is the input to the residual unit, p is max pooling applied to the input, u is the up-sampling, W_i is the weight matrices, σ is the activation function to normalize the output

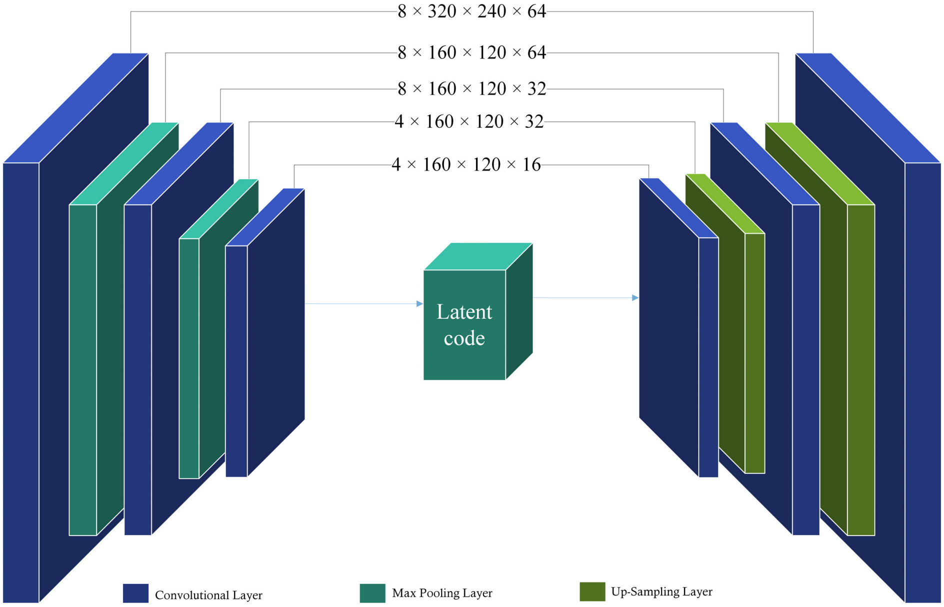

FIGURE 8. Temporal convolution hourglass autoencoder.

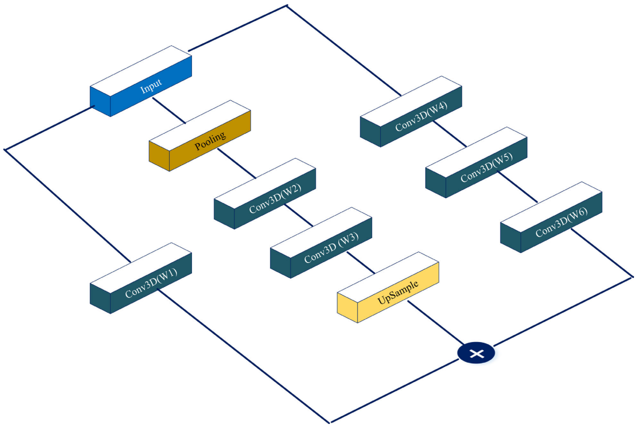

FIGURE 9. Structure of hourglass residual block.

of product z and can be represented using equation(4).

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{4}$$

whearas r is the final output of the residual unit.

The structure of the residual unit r is shown in Figure 9.

3) MAX POOLING LAYER

The max-pooling layer is mainly used to reduce the resolution of the feature maps. It is also called downsampling, where a lower-resolution version of an input signal is created that still contains the large or important structural elements without the fine detail. The pooling layer summarizes the features present in a region of the feature map generated by a convolution layer. Therefore, further operations are performed on summarized features instead of precisely positioned features generated by the convolution layer. This makes the model more robust to variations in the position of the features in the input image.

Our model uses a max-pooling layer in the encoder block, which downsamples the volume by taking the max from each block. The pooling operation involves sliding a three-dimensional filter over a feature map and summarizing the features lying within the region covered by the filter. For a feature map having dimensions $n_h \times n_w \times n_c$, the dimensions of output obtained after a pooling layer is shown in Equation 5:

$$p(x) = \frac{(n_h - f + 1)}{s} \times \frac{(n_w - f + 1)}{s} \times n_c \tag{5}$$

where n_h is the height of the feature map, n_w is the width of the feature map, n_c is the number of channels in the feature map, f is the size of the filter, and s is stride length.

4) DECODER

The Decoder works in a similar way to the encoder, but the other way around. It learns to read the given latent code instead of generating it; based on latent code, latent code reconstructs the input sequence. It aims to minimize the loss while reconstructing. The output is evaluated by comparing the reconstructed frame sequence with the original one. At this point, we propagate backward and then update all the parameters from the decoder to the encoder. Therefore, based on the differences between the input and output images, both the decoder and encoder get evaluated at their jobs and update their parameters for better accuracy. The method uses padded 3D convolutions with stride 2 × 2 × 2, followed by a fixed UpSampling operation for increasing dimensions. In particular, we use 3D Residual Layers in addition to

UpSampling layers, with UpSampling factor 2 × 2 × 2. That is, matrix elements are repeated across each dimension, such that the extent of all dimensions is doubled.

Residual units and upsampling layers are repeated in this manner, resulting in an encoded dimension of 8 × 320 × 240. The set of residual units and upsampling are repeated three times.

E. RECONSTRUCTION ERROR

Latent code is generated in the encoding process based on the input sequence X_i, and the temporal sequence is reconstructed using latent code as shown in equation (6).

$$Y_i = f_d(X_i) \tag{6}$$

Here Y_i is the reconstructed window and f_d is the decoding function that reconstructs the sequence. The temporal sequence contains the number of frames depending on the size of the window W. Frame $X_{(i,k)}$ represents the k^{th} frame in the i^{th} input temporal window X_i, while the $Y_{(i,k)}$ is the k^{th} frame in the reconstructed temporal window. The reconstruction error/anomaly error in our proposed model is calculated based on the pixel-wise reconstruction error. The probability distribution for the input frame pixel $X_{(i,k)}(x, y)$ and the output frame pixel $Y_{(i,k)}(x, y)$ is represented using equation (7).

$$V_{(i,k)}(x, y) = |Y_{(i,k)}(x, y) - X_{(i,k)}(x, y)|^2 \tag{7}$$

where (x, y) is the spatial location of the pixel in the image. The aggregated reconstruction error can be calculated using equation (8).

$$R_{i,k} = \frac{\sum_{(x,y)} V_{(i,k)}(x, y)}{N} \tag{8}$$

where N is the total number of pixel-wise values of reconstruction error and $R_{i,k}$ is the reconstruction error over the k^{th} frame while the frame is present in the i^{th} temporal window. As one frame can show up in multiple windows, as shown in Figure 10. Fr_2 appears in X_1, X_2, and X_3. In this scenario, when a single frame has multiple values of aggregated reconstruction error, then how we will assign a single Abnormality score to a frame. We have defined Abnormality scores at two levels; Window Based Abnormality Score and Frame Level Abnormality Score.

1) IN-WINDOW FRAME LEVEL (IWLF) ABNORMALITY SCORE

In-Window Frame Level (IWLF) is the score that a frame has in its one occurrence in a temporal window. It can be represented as the equation(9)

$$\begin{aligned} IWFL(w, y) &= R_{(w,k)} \\ \exists n(Fr_k, w) &= y \end{aligned} \tag{9}$$

where w is the window number and y is the number of frames in the video. $n(Fr_k, w)$ gives the frame number for the k^{th} frames in the window number w. As we are operating on the binarized image in the model, $IWFL(w, y)$ gives the values in the range (0, 1).

2) CROSS WINDOW FRAME LEVEL (CWFL) ABNORMALITY SCORE

This score represents the mean abnormality score of the frame and can be represented using equation (10).

$$CWFL(y) = \frac{\sum_{w \in L} IWFL(w, y)}{\sum_{w \in L} M(w, y)} \tag{10}$$

where $IWFL(w, y)$ is the In-Window score for the yth frame in window w. $M(w, y)$ is a Boolean function that tells whether a yth frame exists in window w or not.

V. RESULTS AND EVALUATION

A. EXPERIMENTAL SETUP

Proposed methodology is evaluated in the four publicly available data sets, which collected data on normal activities of daily living and falls using visual modalities. As discussed in the Dataset section, five datasets are used for the experimental setup where a camera is installed on the body level of the person being monitored majorly. The fall detection problem in our proposed methodology is approached as an unsupervised problem, which is also considered a one versus all classes problem. For the purpose of experiments, the dataset is processed to be compatible with the one versus all problem. Frames in ADL videos are extracted and stored in a single directory, and similarly, frames in Fall videos are also extracted in a single directory. ADL frames, also termed Normal class frames, are used in the training process, while fall images, also termed Abnormal class, are used in the test process.

B. TRAINING

In this section, data preparation for model training is discussed. Table 2 shows the total data generated in terms of frame sequences for training purposes using equation (2). It is worth noting which frame sequences are generated, Window length (W) of 8, W = 1, and U = 1 is considered(the choice of these parameters is explained in the Methodology Section).

TABLE 2. Frame sequences for the training data.

Dataset	ADL Videos	Frame Sequence	FPS
Thermal Fall Dataset	9	22,063	25 and 15
UR-Fall Dataset	40	8,661	30
SDU Fall Dataset	779	163,579	30
IMVIA Fall Dataset	191	18,975	25
Custom TSH	536	16,884,000	25

The thermal fall data set contains 35 fall videos and 9 videos of normal activities of daily living recorded using the thermal camera. We consider normal activities only for the purpose of training, which after some pre-processing gives

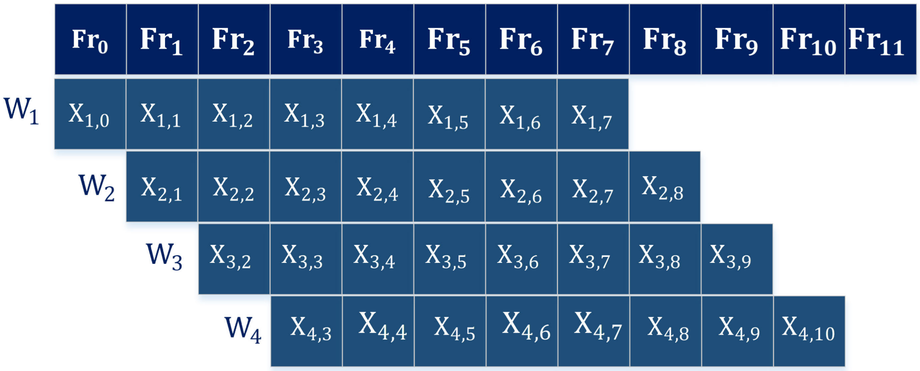
Fr_0 Fr_1 Fr_2 Fr_3 Fr_4 Fr_5 Fr_6 Fr_7 Fr_8 Fr_9 Fr_{10} Fr_{11}
W_1 $X_{1,0}$ $X_{1,1}$ $X_{1,2}$ $X_{1,3}$ $X_{1,4}$ $X_{1,5}$ $X_{1,6}$ $X_{1,7}$
W_2 $X_{2,1}$ $X_{2,2}$ $X_{2,3}$ $X_{2,4}$ $X_{2,5}$ $X_{2,6}$ $X_{2,7}$ $X_{2,8}$
W_3 $X_{3,2}$ $X_{3,3}$ $X_{3,4}$ $X_{3,5}$ $X_{3,6}$ $X_{3,7}$ $X_{3,8}$ $X_{3,9}$
W_4 $X_{4,3}$ $X_{4,4}$ $X_{4,5}$ $X_{4,6}$ $X_{4,7}$ $X_{4,8}$ $X_{4,9}$ $X_{4,10}$

FIGURE 10. Single frame showing up in multiple temporal windows.

22,063 normal frame sequences. UR-Fall dataset contains 40 videos of normal activities of daily living, and 8,661 frame sequences are extracted from the normal activities of daily living. SDU fall dataset contains 997 videos of ADL, which gives 163,579 frame sequences for normal activities of daily living. IMVIA fall dataset contains 191 videos that give 18,975 normal sequences of video frames. Custom dataset contains 342,124 normal sequences, while fall sequences are used from UR-Fall dataset.s

The training data for each dataset is trained on the proposed temporal autoencoder. All experiments are carried out with 300 epochs with the Adam Optimizer as a backpropagation algorithm. The size of the training batch is set to 8, 16, and 32. The value of α is set to 0.01, 0.05, 0.1, 0.2, and 0.5.

Training of multiple models is performed on the windows 4, 8, and 16 changing to vary the receptive field of the model.

C. EVALUATION

All the five datasets have Activities of Daily Living as NORMAL frames and fall frames as ABNORMAL class. In the evaluation stage, only fall frames are considered to calculate the anomaly in the sequence of frames. Each video has at most 16 fall frames in our compilation. In the prediction stage, the predicted labels are calculated using the reconstruction error against the input sequence. Based on the cut-off threshold of the reconstruction error, fall detection/anomaly is detected.

We compare and calculate two methods to calculate the anomaly score in the video sequences.

- In-Window Frame-Level Anamoly Score (IWFL): Based on the reconstruction error of a frame while being part of the window, a frame is labeled fall if the reconstruction error is greater than C_α.
- Cross-Window Frame Level Anamoly Score (CWFL): Based on the average IWFL of a frame, the frame is labeled fall if CWFL is higher than C_β.

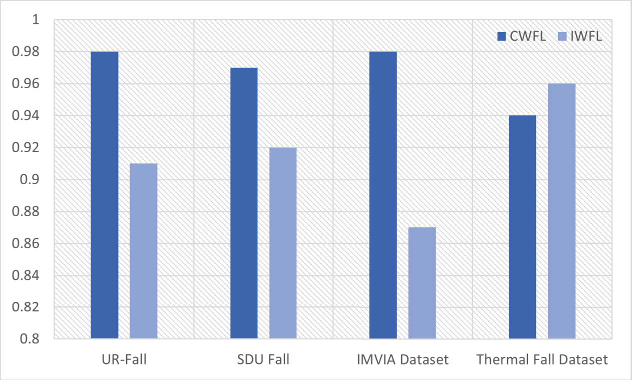

FIGURE 11. Comparison of CWFL and IWFL on the different datasets.

TABLE 3. Comparison of TCHAE with existing algorithms.

	Accuracy	Specificity	Sensitivity	AUC
OneFallGAN [40]	0.92	0.90	0.91	0.89
ARFDNet[41]	0.93	0.91	0.94	0.93
DeepFall[22]	0.95	0.89	0.92	0.92
TCHAE(Ours)	0.98	0.96	0.97	0.96

The threshold of the reconstruction error β defines the predicted label. A value of β greater than 0.55 is considered the anomaly in the image sequence. The evaluation of two reconstruction errors indicates that the cross-window frame level anomaly score is more robust and provides a good measure. Figure 11 gives details of the efficiency of the model on both reconstruction errors. It is clear from the metrics that the cross-window frame level (CWFL) gives a good measure for the fall sequences. CWFL is the average reconstruction error for the particular frame that appears in all fall sequences.

The CWFL for the two fall videos is shown in Figure 12. When a person falls, it is detected as an anomaly, and suddenly reconstruction error spikes, as shown in the graph. An alert is generated for the caregiver to take care of the individual to reduce the damage from falls.

The results are calculated with different configurations, as shown in Table 4. It is obvious from the table that when we apply the skip frames within the window, then the receptive field is increased with fewer windows, and computation can

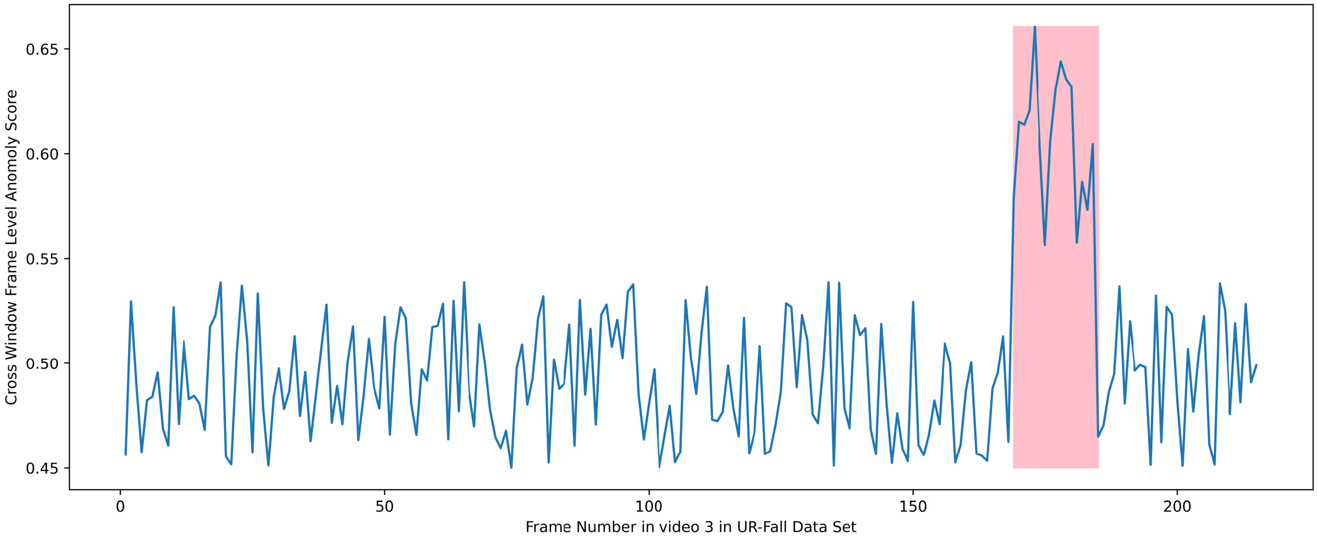

FIGURE 12. Cross window frame level anomaly score.

TABLE 4. Results on different configurations.

Window Length	Frames Gap (U)	Skip Frames (S)	Total Windows	Detection Rate on Frames	AUC	Specificity	Sensitivity
4	1	2	1496	1385	0.90	0.87	0.88
8	1	1	2988	2892	0.96	0.98	0.92
8	2	1	2981	2813	0.93	0.99	0.91
8	2	2	1491	1325	0.91	0.92	0.90
16	1	1	2980	2790	0.82	0.90	0.81

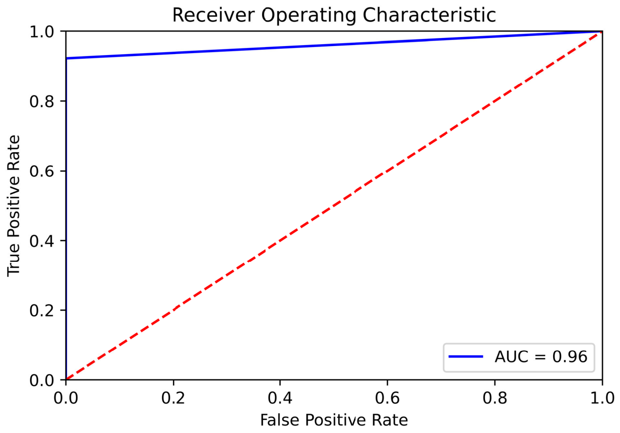

FIGURE 13. Area under the curve for $W = 8$, $U = 1$ and $S = 1$.

be made faster. It is clear that the highest AUC is achieved in the configuration for $W = 8$, $U = 1$, and $S = 1$. Moreover, the In-Window Frame Level score outperforms the Cross-Window Frame Level Score.

D. COMPARISON WITH STATE OF THE ART

The study has been compared with state-of-the-art algorithms. Existing methods are evaluated on the four publicly available datasets, and the average metric for all datasets is reported. It is clear from Table 3 that the Temporal Convolution Hourglass autoencoder outperforms the previous benchmarked algorithms. Although accuracy is not considered to be a good measure due to the high-class imbalance of the fall videos. The reason for reporting accuracy is the reporting of the same measure as in previous studies. We mainly rely on the AUC for the comparison of the existing algorithms.

VI. CONCLUSION

In this paper, we present the solution for the detection of falls in the assistive living environment to solve the open challenges of fall detection, including the rarity of real fall samples, the non-availability of the universal definition of fall and the life-threatening consequences of fatal falls.

The solution is provided using the Temporal Convolutional Hourglass Autoencoder (TCHAE), which is trained on the normal activities of daily living only to solve the problem of data imbalance for the fall samples, and the fall event is treated as an anomaly. On the basis of the reconstruction error, our model shows promising results. We have conducted extensive experiments on the four publicly available benchmarked datasets, including a custom dataset with the compilation of a large set of ADLs. Our proposed solution provides the result in real time by reducing the number of temporal windows with a minor compromise on the accuracy.

Further, the privacy of the person is protected with the use of distributed computing, where the only deidentified stream of the person leaves the environment. The proposed solution has broad application for the detection of anamolous behavior in videos, e.g., seizure detection for epileptic patients and other nursing care anamolies.

VII. AUTHOR CONTRIBUTIONS

All authors contributed to the study conception and design. Data collection, analysis, and experiments were performed by Samyan Qayyum Wahla. The first draft of the manuscript was written by Samyan Qayyum Wahla and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

VIII. COMPLIANCE WITH ETHICAL STANDARDS

- **Ethical Approval** This research problem is approved by the research committee of the University of Engineering and Technology Lahore, Pakistan.
- **Funding Details:** The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.
- **Conflict of Interest** The authors have no relevant financial or non-financial interests to disclose.
- **Informed Consent:** This research is based on publicly available datasets of Fall Detection, reference to datasets are included where the protocols are followed for the consent of subjects participating in research.
- **Data Availability Statement** All used datasets that support the findings of this study are available publically in reference to the dataset section.

REFERENCES

[1] *World Population Ageing 2019: Highlights*. Accessed: May 23, 2022. [Online]. Available: https://www.un.org/en/development/desa/population/publications/pdf/ageing/WorldPopulationAgeing2019-Highlights.pdf

[2] C. Tong, S. A. Tailor, and N. D. Lane, "Are accelerometers for activity recognition a dead-end?" in *Proc. 21st Int. Workshop Mobile Comput. Syst. Appl.*, Mar. 2020, pp. 39–44.

[3] *Falls & Fractures in Nursing Homes How They Happen*. Accessed: Mar. 22, 2022. [Online]. Available: https://www.nursinghomeabusecenter.com/nursing-home-injuries/falls-fractures/

[4] Z. Chen, Y. Wang, and W. Yang, "Video based fall detection using human poses," 2021, *arXiv:2107.14633*.

[5] M. D. Solbach and J. K. Tsotsos, "Vision-based fallen person detection for the elderly," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 1433–1442.

[6] A. Sultana, K. Deb, P. K. Dhar, and T. Koshiba, "Classification of indoor human fall events using deep learning," *Entropy*, vol. 23, no. 3, pp. 1–20, 2021.

[7] W. Chen, Z. Jiang, H. Guo, and X. Ni, "Fall detection based on key points of human-skeleton using OpenPose," *Symmetry*, vol. 12, no. 5, p. 744, May 2020.

[8] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1302–1310.

[9] Z. Guan, S. Li, Y. Cheng, C. Man, W. Mao, N. Wong, and H. Yu, "A video-based fall detection network by spatio-temporal joint-point model on edge devices," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Feb. 2021, pp. 422–427.

[10] M. Musci, D. De Martini, N. Blago, T. Facchinetti, and M. Piastra. (2018). *Online Fall Detection Using Recurrent Neural Networks*. [Online]. Available: https://bitbucket.org/unipv_cvmlab/

[11] G. Santos, P. Endo, K. Monteiro, E. Rocha, I. Silva, and T. Lynn, "Accelerometer-based human fall detection using convolutional neural networks," *Sensors*, vol. 19, no. 7, p. 1644, Apr. 2019.

[12] Y. Ariunbold, S. Brito, and A. Leong, "FallDetectNet: A computer vision platform for fall detection," Stanford, NC, USA, Tech. Rep. 55825567, 2020.

[13] Y. Kong, J. Huang, S. Huang, Z. Wei, and S. Wang, "Learning spatiotemporal representations for human fall detection in surveillance video," *J. Vis. Commun. Image Represent.*, vol. 59, pp. 215–230, Feb. 2019.

[14] S. Chhetri, A. Alsadoon, T. Al-Dala'in, P. W. C. Prasad, T. A. Rashid, and A. Maag, "Deep learning for vision-based fall detection system: Enhanced optical dynamic flow," *Comput. Intell.*, vol. 37, no. 1, pp. 578–595, Feb. 2021.

[15] C. Yao, J. Hu, W. Min, Z. Deng, S. Zou, and W. Min, "A novel real-time fall detection method based on head segmentation and convolutional neural network," *J. Real-Time Image Process.*, vol. 17, no. 6, pp. 1939–1949, Dec. 2020, doi: 10.1007/s11554-020-00982-z.

[16] T. S. Kavya, Y.-M. Jang, E. Tsogtbaatar, and S.-B. Cho, "Fall detection system for elderly people using vision-based analysis," *Sci. Technol.*, vol. 23, no. 1, pp. 69–83, 2020.

[17] F. Harrou, N. Zerrouki, Y. Sun, and A. Houacine, "Vision-based fall detection system for improving safety of elderly people," *IEEE Instrum. Meas. Mag.*, vol. 20, no. 6, pp. 49–55, Dec. 2017.

[18] Y. M. Galvão, J. Ferreira, V. A. Albuquerque, P. Barros, and B. J. T. Fernandes, "A multimodal approach using deep learning for fall detection," *Expert Syst. Appl.*, vol. 168, Apr. 2021, Art. no. 114226, doi: 10.1016/j.eswa.2020.114226.

[19] L. Martínez-Villaseñor, H. Ponce, J. Brieva, E. Moya-Albor, J. Núñez-Martínez, and C. Peñafort-Asturiano, "Up-fall detection dataset: A multimodal approach," *Sensors*, vol. 19, no. 9, p. 1988, 2019.

[20] V. Mehta, A. Dhall, S. Pal, and S. S. Khan, "Motion and region aware adversarial learning for fall detection with thermal imaging," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 6321–6328.

[21] Y. Iguchi, J. H. Lee, and S. Okamoto, "Enhancement of fall detection algorithm using convolutional autoencoder and personalized threshold," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2021, pp. 1–5.

[22] J. Nogas, S. S. Khan, and A. Mihailidis, "DeepFall: Non-invasive fall detection with deep spatio-temporal convolutional autoencoders," *J. Healthcare Informat. Res.*, vol. 4, no. 1, pp. 50–70, Mar. 2020.

[23] X. Cai, S. Li, X. Liu, and G. Han, "Vision-based fall detection with multi-task hourglass convolutional auto-encoder," *IEEE Access*, vol. 8, pp. 44493–44502, 2020.

[24] A. R. Inturi, V. M. Manikandan, and V. Garrapally, "A novel vision-based fall detection scheme using keypoints of human skeleton with long short-term memory network," *Arabian J. Sci. Eng.*, vol. 48, no. 2, pp. 1143–1155, Feb. 2023.

[25] S. Mobsite, N. Alaoui, M. Boulmalf, and M. Ghogho, "Semantic segmentation-based system for fall detection and post-fall posture classification," *Eng. Appl. Artif. Intell.*, vol. 117, Jan. 2023, Art. no. 105616.

[26] A. Khtun and S. G. S. Hossain, "A Fourier domain feature approach for human activity recognition & fall detection," in *Proc. 10th Int. Conf. Signal Process. Integr. Netw. (SPIN)*, Mar. 2023, pp. 40–45.

[27] K. C. Tran, M. Gassi, P. Nehme, J. Rousseau, and J. Meunier, "Video surveillance for near-fall detection at home," in *Proc. IEEE 22nd Int. Conf. Bioinf. Bioeng. (BIBE)*, Nov. 2022, pp. 111–116.

[28] E. García, M. Villar, M. Fáñez, J. R. Villar, E. de la Cal, and S.-B. Cho, "Towards effective detection of elderly falls with CNN-LSTM neural networks," *Neurocomputing*, vol. 500, pp. 231–240, 2022.

[29] D. R. Beddiar, M. Oussalah, and B. Nini, "Fall detection using body geometry and human pose estimation in video sequences," *J. Vis. Commun. Image Represent.*, vol. 82, Jan. 2022, Art. no. 103407.

[30] J. Thummala and S. Pumrin, "Fall detection using motion history image and shape deformation," in *Proc. 8th Int. Electr. Eng. Congr. (iEECON)*, Mar. 2020, pp. 1–4.

[31] Y. Chen, W. Li, L. Wang, J. Hu, and M. Ye, "Vision-based fall event detection in complex background using attention guided bi-directional LSTM," *IEEE Access*, vol. 8, pp. 161337–161348, 2020.

[32] T. Haraldsson, "Real-time vision-based fall detection: With motion history images and convolutional neural networks," Luleå Univ. Technol., Sweden, Tech. Rep. 1254131, 2018.

[33] *UR-Fall Dataset*. Accessed: May 22, 2022. [Online]. Available: http://fenix.univ.rzeszow.pl/~mkepski/ds/uf.html

[34] X. Ma, H. Wang, B. Xue, M. Zhou, B. Ji, and Y. Li, "Depth-based human fall detection via shape features and improved extreme learning machine," *IEEE J. Biomed. Health Informat.*, vol. 18, no. 6, pp. 1915–1922, Nov. 2014.

[35] cy6830mi. (Apr. 2020). *Fall Detection Dataset*. [Online]. Available: https://imvia.u-bourgogne.fr/en/database/fall-detection-dataset-2.html

[36] J. Nogas, S. S. Khan, and A. Mihailidis, "Fall detection from thermal camera using convolutional LSTM autoencoder," in *Proc. 2nd Workshop Aging, Rehabil. Independ. Assist. Living (IJCAI) Workshop*, 2018, pp. 1–5.

[37] R. Dai, S. Das, S. Sharma, L. Minciullo, L. Garattoni, F. Bremond, and G. Francesca, "Toyota smarthome untrimmed: Real-world untrimmed videos for activity detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 2533–2550, Feb. 2023.

[38] A. Kirillov, Y. Wu, K. He, and R. Girshick, "PointRend: Image segmentation as rendering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9796–9805.

[39] K. Schindler and L. van Gool, "Action snippets: How many frames does human action recognition require?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[40] Y. M. Galvão, L. Portela, P. Barros, R. A. de Araújo Fagundes, and B. J. T. Fernandes, "OneFall-GAN: A one-class GAN framework applied to fall detection," *Eng. Sci. Technol., Int. J.*, vol. 35, Nov. 2022, Art. no. 101227.

[41] S. K. Yadav, A. Luthra, K. Tiwari, H. M. Pandey, and S. A. Akbar, "ARFDNet: An efficient activity recognition & fall detection system using latent feature pooling," *Knowl.-Based Syst.*, vol. 239, Mar. 2022, Art. no. 107948.

SAMYAN QAYYUM WAHLA was born in Narowal, Pakistan. He received the bachelor's degree in computer science from the University of Engineering and Technology Lahore, Pakistan, in June 2014, the master's degree in computer science, in April 2017, and the Fazil-e-Tibb-wal-Jarahat (FTJ) i.e., Diploma in Unani Medical System and Surgery (DUMS) degree, in 2021. He is currently pursuing the Ph.D. degree in computer science with the University of Engineering and Technology Lahore. His work experience includes full stack software development with different software companies and as a Research Team Lead with the Computer Vision and Machine Learning Laboratory. He is a Lecturer with the University of Engineering and Technology Lahore. His research interests include AI powered ERP solutions, healthcare automation, machine learning, computer vision, and deep learning.

MUHAMMAD USMAN GHANI received the Ph.D. degree from The University of Sheffield, U.K. He is currently a Professor with the Department of Computer Science, University of Engineering and Technology Lahore, Pakistan. His Ph.D. study was concerned with statistical modeling for machine vision signals, specifically language descriptions of video streams. He has been studying on spoken language processing using statistical approaches with applications, such as information extraction from speech and speech summarization. His recent work is concerned with multimedia, incorporating text, audio, and visual processing into one framework. He has over 15 years of research experience specifically in the areas of image processing, computer vision, bioinformatics, medical imaging, computational linguistics, and machine learning.