Bachelor of Science in Computer Science & Engineering

Vision-Based Indoor Human Fall Event Detection Using Deep Neural Network

by

Chumui Tripura

ID: 1604131

Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)

Chattogram-4349, Bangladesh.

September, 2022

Vision-Based Indoor Human Fall Event Detection Using Deep Neural Network

Submitted in partial fulfilment of the requirements for
Degree of Bachelor of Science
in Computer Science & Engineering

by

Chumui Tripura

ID: 1604131

Supervised by

Dr. Kaushik Deb

Professor

Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)

Chattogram-4349, Bangladesh.

The thesis titled '**Vision-Based Indoor Human Fall Event Detection Using Deep Neural Network'** submitted by ID: 1604131, Session 2019-2020 has been accepted as satisfactory in fulfilment of the requirement for the degree of Bachelor of Science in Computer Science & Engineering to be awarded by the Chittagong University of Engineering & Technology (CUET).

Board of Examiners

Chairman

Dr. Kaushik Deb
Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Member (Ex-Officio)

Dr. Md Mokammel Haque
Professor & Head
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Member (External)

Muhammad Kamal Hossen
Associate Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Declaration of Originality

This is to certify that I am the sole author of this thesis and that neither any part of this thesis nor the whole of the thesis has been submitted for a degree to any other institution.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. I am also aware that if any infringement of anyone's copyright is found, whether intentional or otherwise, I may be subject to legal and disciplinary action determined by Dept. of CSE, CUET.

I hereby assign every rights in the copyright of this thesis work to Dept. of CSE, CUET, who shall be the owner of the copyright of this work and any reproduction or use in any form or by any means whatsoever is prohibited without the consent of Dept. of CSE, CUET.

Signature of the candidate

Date:

Acknowledgements

First and foremost, I am indebted to the Almighty for continuously giving me patience, knowledge, and strength. I also want to convey my gratitude to all the well wishers who has supported and encouraged me to go on forward on completing my thesis work.

Secondly, I would like to convey my immense gratefulness to my supervisor, Dr. Kaushik Deb, Professor, Department of CSE, CUET, for his endless guidance, motivation, and support. It would be very difficult for me on the journey of my thesis if it wasn't for his persistent supervision. His intellectual suggestions and analytical questions has helped me to broaden my horizon and see in alternative perspectives. I am grateful for the inspiration and courage that he always conducted during the sessions.

I would also like to express my thankfulness to the board members for giving me their valuable feedback, and advice. I am sincerely grateful to the faculty members for providing me with knowledge and wisdom throughout my undergrad journey.

And lastly, I want to show my thankfulness to my parents for their unconditional support and love. It wouldn't be possible for me to keep my spirit up without their blessings and prayers.

Abstract

In current times, the rate of lonely individuals living is increasing swiftly. And among them, the number of older people is the most. The older people often suffer from many diseases because of their age and they also lose mobility. As a result, the fall incident happens quite often times which leads to serious physical injuries. And if they are not treated in time they may suffer from long-term fatal disease or even worse death. Therefore, keeping their sufferings in mind, we have come up with a vision-based deep learning technique for fall event detection. Because surveillance system has become very cheap and available nowadays. To build the model, at first, we extracted the keyframes from video data by incorporating the K-means clustering. After that, segmentation technique have been applied with Mask-RCNN model. We used the Mask-CNN model for person detecting and segmenting which also handles the partial occlusion case. Further, spatial features are derived with the help of convolutional neural network (CNN) that is the pre-trained VGG16 in our case. Also, for handling the temporal dependency of the frames, bi-directional gated recurrent unit(Bi-GRU) have been incorporated. And lastly, the softmax classifier outputs our desired result. The publicly available U-R Fall Dataset and Multiple Cameras Fall Dataset have been used for our experiment. We also evaluated the model with our self-built dataset. However, our proposed model has shown 100% accuracy for the U-R Fall Dataset, and 99.4% accuracy for the Multiple Cameras Fall Dataset.

Keywords: Fall detection; vision-based; deep learning; K-means clustering; Mask-RCNN; pre-trained VGG16; bi-directional gated recurrent unit(GRU).

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

A "fall" is defined as an unwanted incident that results in when someone comes to rest on the ground, a floor, or another lower level surface that causes serious damage to them.. Falls have become a crucial health concern, especially among elderly people. According to a World Health Organization report, 28% - 35% of persons aged above 65 die each year, and the percentage rises to 32%- 42% for people aged 70 and up [1].

In current years, statistics have revealed that human falls are a leading cause of harmful death among elderly individuals aged over 79, both in Bangladesh and in Western countries [2]. The World Health Organization (WHO) has stated that by 2050, there would be 20 billion people worldwide who are 60 years of age or older, growing from 900 million in 2015. This represents 22% of the world's inhabitants. [3].

According to statistics, 93% of the elderly live in private residences, with 29% of them living alone [4]. Even without external injuries, 50% of the seniors who lay on the ground for more than an hour after a fall died around six months after the injury [5]. The length of the hospital stay might range from four days to the rest of one's life, depending on the patient's level of vulnerability. Falls can lead to sadness, diminished independence, reliance from being immobilized, and many other limitations on everyday activities. The effects of falls were identified in a survey of elderly falls reported in [6], as seen in figure 1.1.

Therefore, there is a critical need to design an intelligent fall detection method for the elderly that can detect falls automatically and instantly and alert caretakers or family members.A fall detection method might improve the lives of seniors who

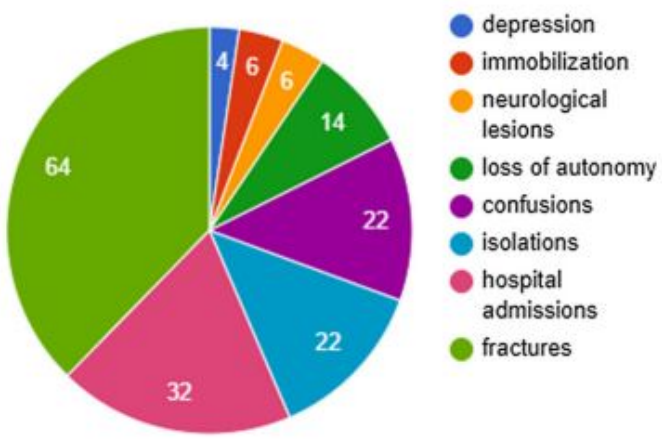


Figure 1.1: Consequences of fall [7]

live alone. It will not only just help them to grow confidence in free movement but also assist the caretakers and medical institutes to monitor conveniently.

1.2 Background

Many methods have been proposed lately based on modern devices to detect fall events. Fall detection algorithms can be divided into three categories: computer vision-based methods, ambiance sensor-based, and wearable sensor-based [8]. In wearable sensor-based methods, the motion status or location information is determined by devices worn on the older person's body. Though wearable devices have shown effectiveness in detecting falls yet, their limitations such as battery lifetime, being forgotten to wear or disconnection have hindered their usage. Not just that, wearable sensors involve precise positioning which is difficult to acquire.

On the other hand, ambiance sensor-based systems include audio, pressure, and vibration sensors. And these sensors are placed across the surroundings, which are usually indoors. Pressure, Wi-Fi, infrared arrays, audio, vibration, radar, and other signals are employed for detection. The fundamental idea behind this type of fall detection technology is to use wireless techniques to detect changes in the environment and to establish a link between wireless transmission and human actions. But the vibration and audio sensors might produce false alarms from falling objects. And the cost of this technology is high as well. Therefore, it limits their scope too. As a result, the employment of computer vision-based devices is

becoming more popular for higher precision and robustness.

There exist several algorithms to effectively detect fall events in computer vision-based technology. And it does the task by using the video or image data acquired by the surveillance system or digital camera, which is very cost-effective. Not just that, the method processes the data in a very fast manner as well. Therefore, the real-time output can be get as well if steps are followed properly.

1.3 Framework/Design Overview

Among many methods of fall detection we decided to work with the vision-based deep learning technique for it's simplicity and acceptance. Thus, we will be working with video data acquired through cameras. And there will be some preliminary steps to follow if we intend to process video data for our deep learning models. The block diagram of our proposed framework is presented in figure 1.2 . Our framework comprises of the following main steps:

Figure 1.2: Block diagram of our fall detection framework

- Input videos: We will be dealing with video data in order to find the fall or non-fall event.
- Data Pre-processing: It's necessary that one should process the raw data for further analysis as it will not only reduce model's computation cost but also help to increase model's performance
- Spatial Feature Extraction: Most of the deep learning approach uses the CNN network for extracting the spatial features. And spatial features are the important property to recognize, differentiate, or identify a specific object of an image.
- Temporal Feature Extraction: Activities in a video data changes with time.

And this temporal dependency can be dealt with sequence models of deep learning approach.

1.4 Difficulties

To implement the model we faced several issues. And it was quite challenging for us to handle those issues. The challenges that faced are given below:

Occlusion: Although most of the data are captured in a simulated environment, there might exist some furniture in the scenes. And these furniture had caused some occlusion problem.

Scarcity of datasets: Deep learning models require massive amount of data to properly predict an unknown data. But datasets on human fall events are very scarce.

Uneven length of videos: The available datasets come with uneven lengths of videos. Therefore, finding the keyframes from each video is quite challenging.

Similarity of events: The fall event and lying event are quite similar. Therefore, the model can get confused to properly distinguishing between them

1.5 Applications

Although we are building our model for assisting the alone elder individuals, our method can be utilized in various applications as well. Our proposed method is applicable in the following areas:

- It can be utilized as the alert system in the medical application. If fall accidents can be detected properly immediate medical assistance can be provided. As a result, it may save the injured person from further critical medical conditions.
- It can be used as an assistive technology in monitoring the individuals in the nursing home. Keeping observation at several people might become difficult for the person in charge. This technique will be of great use in that case.

1.6 Motivation

Elderly falls almost often result in major health problems, as well as a loss of physical fitness. The most common damage in an elderly person's fall is a fracture, but there's also a chance of brain trauma, coma, and paralysis. Immediate medical assistance can decrease the fatality of any kind of physical damage or injuries. According to World Health Organization (WHO), it is considered the second leading cause of unintentional injury death worldwide, we can see how severe this problem is. Thus, there is no telling of the importance of the fall detection system.

It is critical to implement a comprehensive monitoring system for senior citizens that can identify fall actions inside the room quickly and automatically in order to reduce after-fall impacts. As a result, an effective approach is provided here for automatically classifying fall incidents from regular lives by incorporating the video data. The primary motivation behind our thesis work can be listed as below:

- **To overcome existing limitations:** Most existing work suffer from high computation cost and dataset variation. Our goal is to reduce the training time and explore with variation of dataset.
- **To handle occlusion issue:** As existing dataset include very few occlusion data, we will experiment with artificially generated partial occlusion scenario.
- **To evaluate model for outdoor environment:** Most of the existing datasets were prepared inside indoor environment. So, our objective will be to evaluate our model under natural outdoor light.
- **To come up with key frame selection technique:** Key frames are the important part of building a model as they summarizes the whole video. Therefore, it's necessary to follow the techniques that can generate the most relevant key frames.

1.7 Contribution of the thesis

The contribution of our proposed model is given in the following:

- The prime contribution of our work is handling occlusion. We introduced the pre-trained Mask R-CNN model for segmenting the detected person. And this model can even segment the person in partial occlusion situation.
- We evaluated our model with self-built dataset that was performed under natural light on outdoor environment.

1.8 Thesis Organization

The organization of the rest of the thesis report is as follows:
Chapter-2 presents a brief analysis of previous works related to our thesis as well as its framework **Chapter-3** provides the general steps of our methodology and the experimental design **Chapter-4** describes the evaluation, experimental results, as well as comparisons of our work **Chapter-5** summarizes the findings of our thesis work

1.9 Conclusion

In this chapter, we have briefly described our research work that have been carried out. The motivation behind our work as well as the difficulties that we faced have been discussed. A brief overview of our framework along with the contribution have been presented as well. In the following chapter, current state and literature review relating to our task shall be provided.

Chapter 2

Literature Review

2.1 Introduction

In this chapter, we tried to discuss the different existing research works that classify the human fall event. Most of the previous works have incorporated the use of Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) models. And these models have proven to give the best results in terms of classification. Therefore, the architecture of both the CNN and RNN is explained here. But as we are working with video data we need to learn about the key-frame extraction method first. This section further provides a brief explanation of some of the previous related works.

2.2 Key-frame extraction

A video sequence is made up of many frames, which are still images. Keyframes are those events in the video that are particularly important. One of the crucial tasks in video data processing is key-frame extraction. Because these keyframes summarize the whole video. And important features of the videos can be get from these key-frames which is necessary for further processing. However, there are many approaches for key-frame extraction. Some of the well-known approaches are sampling-based, shot-based, clustering-based and so on [9]. An illustration of keyframe extraction process is given in figure 2.1

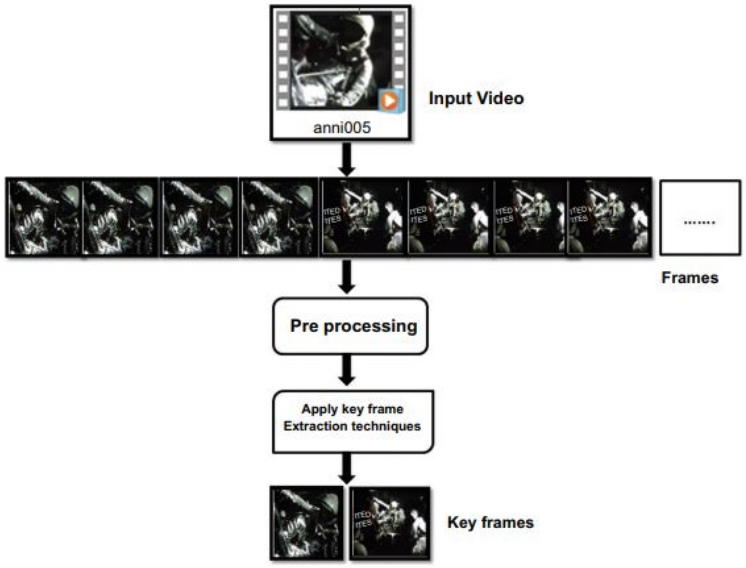


Figure 2.1: Key-frame extraction

2.3 Segmentation of Target Object

The process of grouping together portions of an image that correspond to the identical object class is known as picture segmentation. This method is commonly named as "pixel-level classification." To put it another way, it implies dividing up images (or video frames) into several items or segments. However, it's been found that segmenting the target object can handle the occlusion problem [10] [11]. Therefore, to solve one of our challenge such as the occlusion issue, we can implement the available latest segmentation method.

2.4 Convolutional Neural Network

Convolutional Neural Networks function by taking an image, giving it a weight depending on different features in the image, and then differentiating between them. In comparison to other deep learning algorithms, CNN needs very less pre-processing of the data. One of CNN's key strengths is that it trains its classifiers using simple techniques, which enables it to learn the properties of the target object. The CNN is made up of three different kinds of layers: convolutional layers , pooling layer, and fully-connected (FC) layer. A CNN architecture is created when these layers are overlaid. A brief explanation of the CNN architecture is

given below:

2.4.1 Convolution layer

This is the first layer that is utilized to extract the different features from the input pictures. Mathematical operation is performed between the input image and kernel in this layer. As a result, it produces a convolved feature. Typically, the input image's shallow features are extracted by this layer. The convolved feature may provide us details about the image such as the edges and corners. Once the convolution operation has been applied to the input, CNN's convolution layer transfers the output to the following layer. Figure 2.2 illustrates the colvolution operation of CNN.

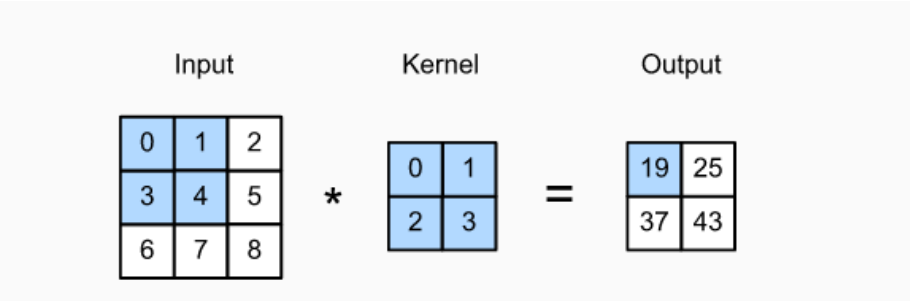


Figure 2.2: Convolution operation

2.4.2 Pooling layer

A pooling layer often comes after a convolutional Layer. The main objective of this layer is to reduce the size of the feature map in order to decrease the computational cost. Essentially, it is a summary of the features produced by a convolution layer. However, there are mainly three types of pooling operation such as max pooling, sum pooling, and average pooling. In max pooling, the largest element of the feature map is considered. Sum Pooling computes the total sum of the values in the specified section. In average pooling, the average value of the feature map is taken into consideration. Among these, the max pooling operation is incorporated mostly. The max pooling operation is illustrated in figure 2.3.

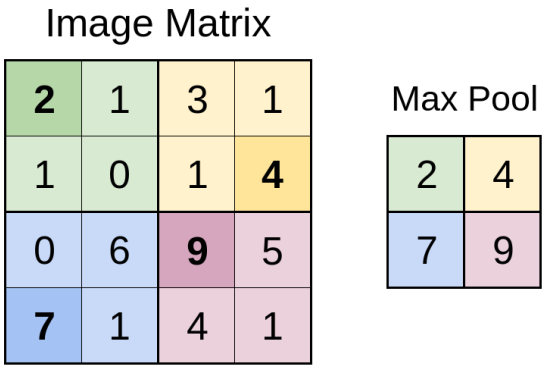


Figure 2.3: Max pooling operation

2.4.3 Fully connected layer

The fully connected (FC) layer, which links the neurons within two layers, comprises of weights and biases. These layers make up the final few layers of a CNN architecture. This process flattens the input image from the preceding layers and feeds it to the FC layer. The flattened vector is subsequently subjected to a few more FC layers, which are often where mathematical function operations happen. At this point, the classification process gets started.

2.5 Recurrent Neural Network

Recurrent neural networks (RNNs) are unique types of feed-forward neural networks with a specialization in temporal domain modeling. The ability of RNNs to transmit data over time steps makes them distinctive. In order to facilitate training in the temporal domain and take advantage of the input's sequential nature, RNNs feature an additional parameter matrix for interconnections between time steps as part of their structural design. RNNs are trained to produce results where predictions on every time step are based on both the most recent input and data from earlier time steps. RNNs can be used to analyze time series-related input. Anyway, this model suffers from the vanishing gradient problem. The architecture of RNN is illustrated in figure 2.4.

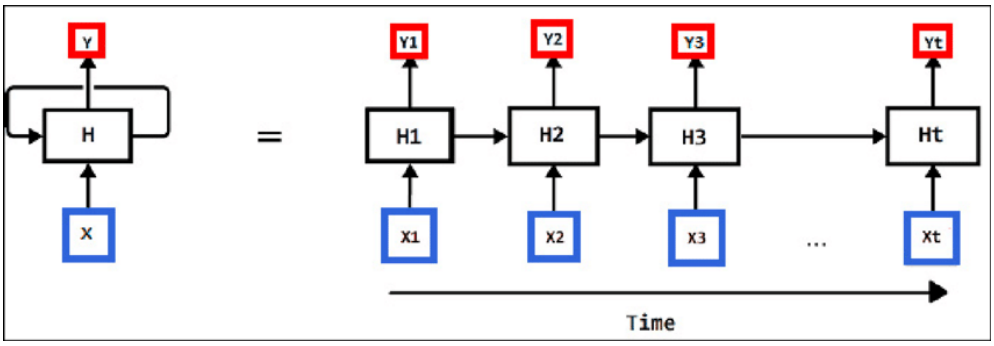


Figure 2.4: Architecture of the Recurrent Neural Network (RNN)

2.6 Related Works

The improvement of modern healthcare and medical facilities is a comprehensive study topic that is highly sought after. A lot of attention has been paid to fall prevention in particular since falls among the elderly can have serious medical and psychological effects. This section briefly discusses the works related to fall detection systems and their limitations.

In [12], Adhikari et al. proposed a method of activity recognition for a fall detection system by incorporating the Depth image and RGB image of the Kinect sensor. At first, they applied the background subtraction method to process the images. After that, to analyze and classify the characteristics of the activity, they used the CNN model which then identifies the lying pose as a fall event. Their approach has achieved 74% of accuracy on test data. However, the result could be improved more if Long-Short-Term-Memory (LSTM) is applied just as in [13].

Abobakr et al. focused on classifying the fall events and normal activities of daily living(ADL) by introducing an end-to-end deep learning architecture[13]. The architecture is composed of ResNet-17 and Long-Short-Term-Memory (LSTM). They used the Depth information from video frames acquired by Microsoft Kinect sensor which preserves privacy and even works in no light environment. Their method has given 98% accuracy on the validation set which is better than accelerometer-based systems. Also, they claim to provide a real-time fall detection alarm with their approach. But the major lacking of their work is, they experimented with their method with only a single dataset and they hadn't shown whether it would give the same accuracy with other datasets or not.

Instead of using traditional 2D-CNN architectures, Lu et al. [14] have utilized 3D CNN to extract the spatial as well as the motion features from kinematic video frames. And further to classify the fall event from sequential data an attention-based LSTM has been incorporated. As a result, this model is capable of operating in dynamically changing visual circumstances. However, the major drawback of their approach is the high computational cost. Yet, this method has shown superiority even with a small fall dataset.

In [15], a model for describing the direction of human movement depending on skeleton extraction is presented utilizing spine ratio and deflection angles. However, due to a small dataset, this model misclassifies in several situations, such as in exercise motion.

Paper [16] takes into account deformation angles with spine ratio when estimating human posture from the skeleton. To extract the human skeleton, it uses the OpenPose algorithm. An algorithm for time-continuous recognition has been designed for action classification. They did, however, suggest that in the future, their suggested methodology might be combined with a deep learning model.

In [8], the deep learning-based background subtraction method Mask R-CNN is applied to detect a person. After that to extract the visual features VGG-16 is incorporated. Finally, the attention-guided Bi-directional LSTM is used to classify the fall event. Their proposed approach has shown significant accuracy than some state-of-the-art methods which is 96.7%. However, their major drawback is, their method works very well with the solitary scene but they haven't stated whether it will provide the same accuracy if there's more than one person present in the scene.

Xi cai et al. in [17] has proposed a fall detection method based on the dense block with a multi-channel convolutional fusion (MCCF) strategy. The MCCF-DenseBlock can extract the information thoroughly and the accompanying transition layer helps to lessen the data accumulation. However, their approach has achieved an F-score of 0.973 which outperforms several state-of-the-art methods. But, their major drawback is they experimented with only a single dataset which limits their scope of usefulness.

To experiment with the complex scenes Feng et al. in [18] has proposed a new dataset with multiple person in it. They further introduced an attention guided LSTM model to classify the fall event from the video data. They also incorporated the Deep-Sort method to keep track of the target person from other pedestrians. And for the spatial feature extraction they used the VGG166 model which will be further used for the fall event prediction. However, their experiment result is not much very promising from other state-of-the-art methods.

In [19] Min et al. suggested a novel approach based on deep learning and activity features for scene analysis to identify human falls on furniture. The suggested solution first analyzes the scene using the Faster R-CNN deep learning algorithm to find people and objects like sofas. They experimented on their self-collected data and benchmark dataset UR fall detection dataset. And their approach gave 95.50% accuracy, 94.95% recall, and 94.44% precision. However, they didn't mentioned how their method will handle the temporal dependency of videos.

Human pose is extracted using the Fast Pose Estimation method which is further used for human fall detection in [20]. They stored the extracted pose location information which is then transfered to both their 1D-CNN and Time-distributed CNN-LSTM model. These models classify the fall and non-fall event.

In [21] indoor human fall classification is done using CNN in combination with gated recurrent unit (GRU) network. They built their model from scratch where the CNN model has nine convolution layers and GRU consists of one layer. Their srcatch model gave 99.8% of accuracy on UR fall detection dataset and 98% accuracy for Multiple cameras fall dataset. They also experimented with other transfer learning models like VGG16, VGG19, Xception etc. And, their scratch model outperformed many other existing methods. Although, their work comes with limitations such as high computation cost, and illumination effect.

Anitha et al [22] discussed a fall detection approach established on deep learning method that involves video preprocessing, spatial and temporal feature extraction, and classification with stacked auto-encoder. The MobileNet is used for the spatial feature extraction and GRU in employed for solving the temporal dependencies. They incorporated the group teaching optimization algorithm (GTOA)

for optimizing and fine tuning their model. However, they didn't mention whether their model could handle occlusion or shadow issues.

Using the displacement of spatial features of human body while fall and non-fall activity is considered for the binary classification in [23]. They employed both the support vector machine(SVM) and K-nearest neighbor(KNN) for classifying the activity. Height-to-width ratio, displacement in the vertical and horizontal centroid movement are considered as the key feature for fall events. Anyway, their work can't handle multiple person environment and natural light issues as there is no publicly available datasets for that.

2.7 Conclusion

In this chapter, summary of related works has been explained. We can see in most of the work CNN has been introduced for spatial feature extraction. After getting the spatial features to maintain the temporal sequence RNN has been incorporated. Although there are many other approaches to detect human fall event. Our work mainly focuses on vision-based deep-learning approaches. In the next section, we will discuss our proposed methodology in detail.

2.7.1 Implementation Challenges

The implementation challenge of our thesis work is given below:

- **Difficulty to deal with small dataset:** Usually deep learning model requires a lot of data to get trained on for better performance. The existing available datasets for fall detection comprises of very small number of videos. The U-R fall dataset consists of 70 videos, whereas the Multiple Cameras fall dataset comprises of 192 videos. Therefore, it's pretty challenging to deal with this small amount of data.
- **Difficulty in handling occlusion:** For indoor environment occlusion by household furnitures are very common. There are only a few occlusion data in Multiple Cameras fall dataset. So, to verify, whether our model can

properly handle occlusion we need quite amount of occlusion data. As a result, it's become a challenge to evaluate our model.

- **Difficulty in key-frame extraction:** Most of the existing work either didn't specify extraction of key-frames or did it by skipping frames of interval. So, to choose the most relevant frames from a video is quite challenging.

Chapter 3

Methodology

3.1 Introduction

In this chapter we will discuss the framework of our task in detail. As we will deal with the video data, our method will start by taking video as input and end by classifying between fall and non-fall event. Our model will perform as binary classifier. There will be several steps to be followed in our methodology.

The workflow will be start by acquiring video data from a camera or surveillance system. After that, we need to prepare the video data for our model. And it will be done in the data pre-processing step. We will begin by selecting the important frames also known as the key frames from the video. We will perform the key frame extraction using the clustering method and we considered the K-means clustering in our task. After that, for handling occlusion issue we will use the Mask R-CNN for segmenting our target person. Mask-RCNN is used for both person detection and segmentation task. Frames are then resized to be compatible with the model and for lessening computational cost.

Spatial feature extraction will be done applying the pre-trained VGG16 model. After extracting the spatial features it will then passed upon to the sequence model. And temporal feature will be extracted using the bi-directional GRU. This will solve the long-term-dependencies of our data. Finally, the softmax function will predict the fall/non-fall event from the video data. After building the model, the parameters are tuned for optimization. Figure 3.1 illustrates our proposed methodology.

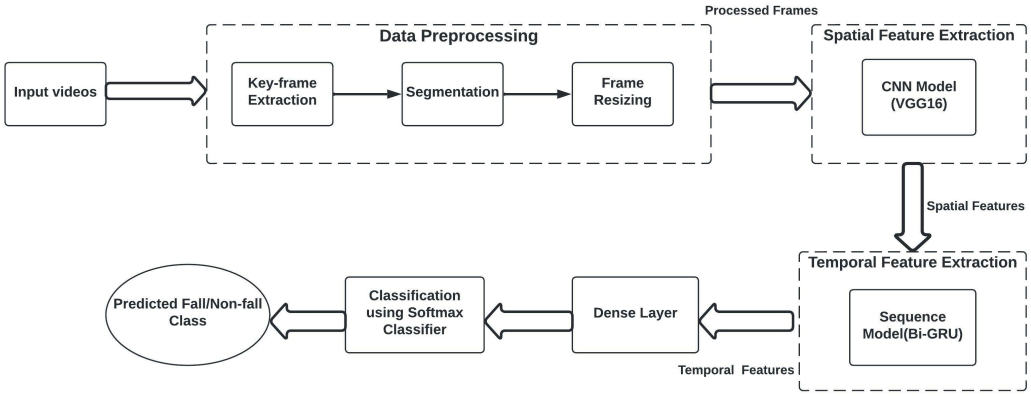


Figure 3.1: Workflow of our proposed methodology

3.2 Dataset Description

We have experimented our model on two datasets. They are the high quality dataset UR fall detection dataset and Multiple cameras fall dataset (MCFD).

3.2.1 UR Fall Detection Dataset

There are 70 videos data in the UR fall detection dataset. The dataset contain 30 videos of fall event and 40 videos of Activities of Daily Living (ADL). The activities of daily living (ADL) actions were captured with just one Microsoft Kinect camera and one accelerometer, whereas the fall actions were acquired by two Microsoft Kinect camera and one accelerometer. For fall identification in this situation, we selected the RGB videos in the UR Fall Dataset that were recorded by camera 0. All the videos has a frame rate of 30 fps with a resolution of 640×240. Some samples of RGB video frames of URFD dataset is illustrated in figure 3.2.

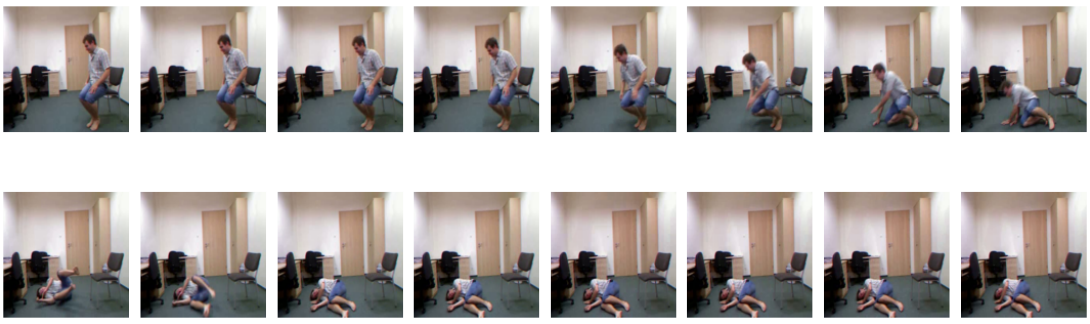


Figure 3.2: Sample RBG frames of a video of URFD dataset

3.2.2 Multiple Cameras Fall Dataset

The Multiple Cameras Fall Dataset is another well-liked dataset utilized in experiments on fall detection. This dataset includes 24 scenarios, each of which has nine possible outcomes, such as walking, falling, lying on the ground, crouching, and so forth. Eight cameras were used to record each scenario separately, yielding 192 videos with a total of 736 actions.Each frame is 720x480 pixels, and each video comes with a frame rate of 30 fps. The illustration of sample frames of a video from MCFD dataset is given in figure 3.3.

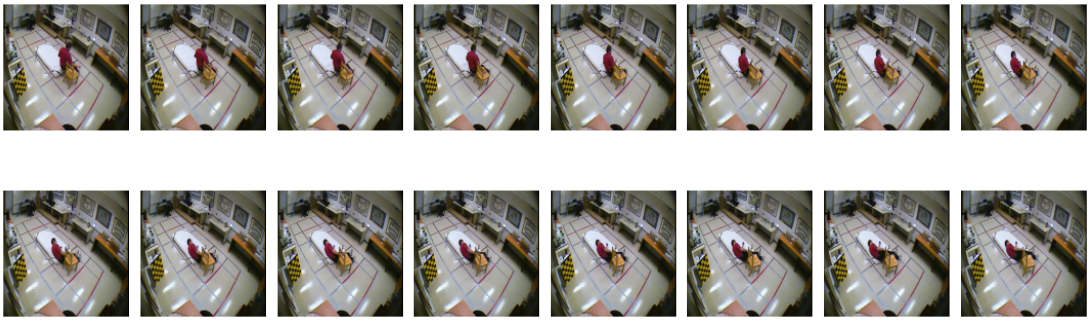


Figure 3.3: Sample RBG frames of a video of MCFD dataset

3.3 Detail Explanation

The detail explanation of our step by step process for the fall detection classification is given in the following:

3.3.1 Data Preprocessing

It's one of the crucial step for any machine learning or deep learning approach. Because in this step the raw data is prepared to be compatible for building the model. Data preprocessing depends on the requirement of the task. And this step is usually done at the beginning of any machine or deep learning task. We have preprocessed our video data by the following steps:

3.3.1.1 Key Frame Extraction from Video

Extracting the key frames from video is one of the most important part. Because all the following steps will depend on these key frames. Therefore, it's a crucial step in our methodology. However, we have incorporated the clustering method for choosing the key frames. We selected 16 frames that summarizes the whole video. Some of the image features such as RGB color histogram, and Histogram of Oriented Gradients (HOG) features have been considered for clustering the frames. We chose the K-means clustering for the key frame selection where K indicates the number of key frames which is 16 in our case. The flowchart of our key frame extraction method is given in figure 3.4

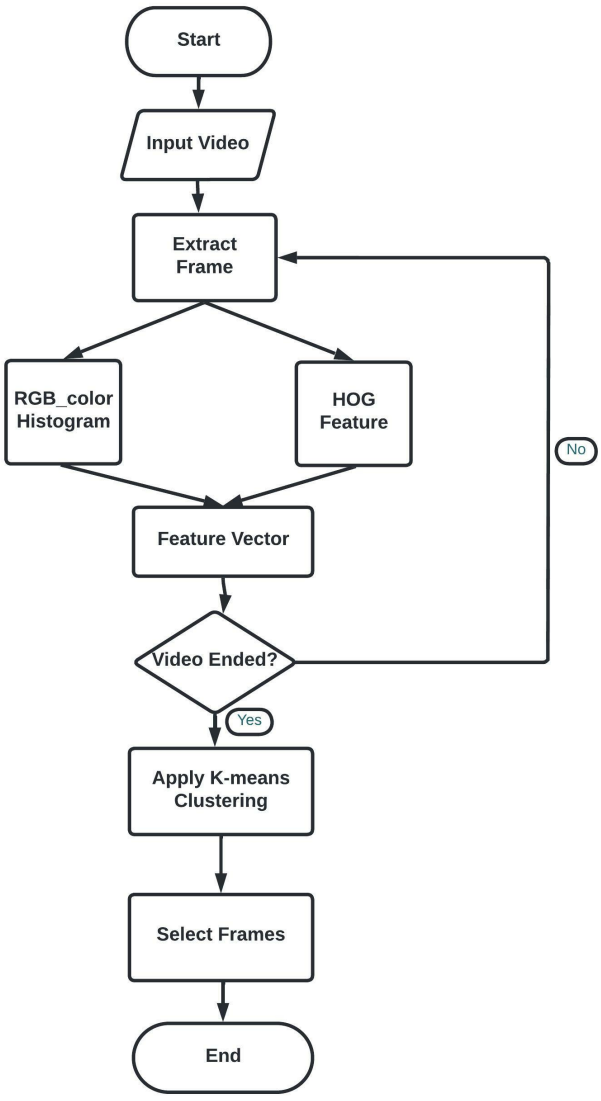


Figure 3.4: Flow chart of our key-frame extraction method

3.3.1.2 Segmenting target object

To handle occlusion we have considered the segmentation method. It will not only handle occlusion issue but also will act as an background subtractor to reduce the computaional complexity. There are many deep learning based approach for image segmentation. However, we picked the Mask R-CNN segmentation model for our task.

Mask-RCNN is known as the state-of-the-art in image segmentation model. And our Mask-RCNN model is already pretrained with COCO dataset which is an extensive dataset for object detection, segmentation, and captioning. It can also act as person detector and background subtraction model. It can detect and segment the human body contour even during the occlusion incident and subtracts the background. Thus,it gives region of interest of the target object which will be further used for feature extraction. The architecture of Mask-RCNN consists of a backbone network (ResNet101+Feature Pyramid Network), Region Proposal Network(RPN), and Mask representation. It returns candidate object, class, bounding box and object mask. The architecture of Mask-RCNN model is illustrated in figure 3.5.

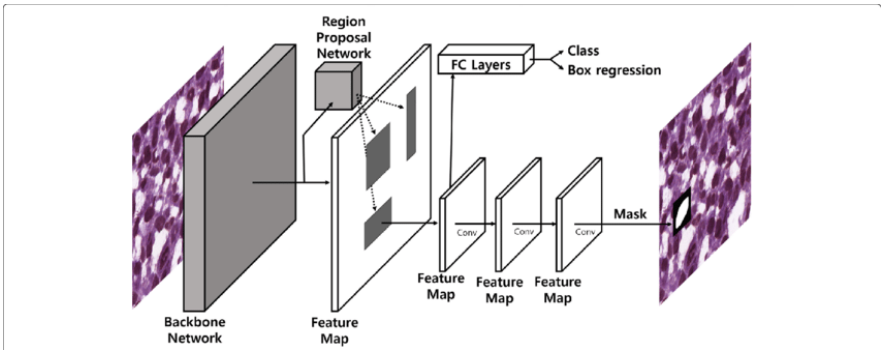


Figure 3.5: Architecture of Mask-RCNN

3.3.1.3 Frame Resizing

To achieve the desired result, it is important to choose a size for frames before extracting the features. It is commonly used to both simplify and improve the accuracy of models. And we have performed the frame preprocessing by resizing of the frames. However, the datasets that we have considered for our experiment

comprises of different sizes of frames. This may lead to increase computational cost as well as reduce model's efficiency. Therefore, we have resized all the frames to a fixed size which is 224×224 in our case.

3.3.2 Spatial Feature Extraction

After following the preprocessing steps, we will need to extract the spatial features from our data. And to do the task CNN model has been employed. There are many pre-trained CNN models in existence. These models perform great in extracting the spatial features. We employed the VGG16 model which is one of the popular pre-trained model for feature extraction. As we will be using the model for feature extraction only, we will exclude the output layer of the model which is the fully connected layers. By doing so we will get the feature stack from the model which will be further used for temporal feature extraction. Below we will describe the architecture of VGG16 model in detail.

3.3.2.1 Architecture of VGG16

VGG16 is an object identification and classification method that has a 92.7% accuracy rate when classifying 1000 images into 1000 separate classes. It is a well-liked technique for classifying images and is simple to employ with transfer learning. VGG16 is comprised of 13 convolutional layers, 3 fully connected layers and 5 max-pooling layers. Consequently, there are 16 layers with tunable properties (13 convolutional layers and 3 fully connected layers). The model's name, VGG16, was chosen for this purpose. The convolutional layers in VGG-16 are all 3×3 filter with a stride size of 1 and the same padding. On the other hand, the max pooling layers are all 2×2 filter with a stride size of 2. Throughout the entire architecture, convolution and max pool layers are arranged in the same manner. In figure 3.6 the architecture of VGG16 is given. Also, as we are taking only the feature stack from the model, the fully connected layer that outputs the 1000 class are removed from the model.

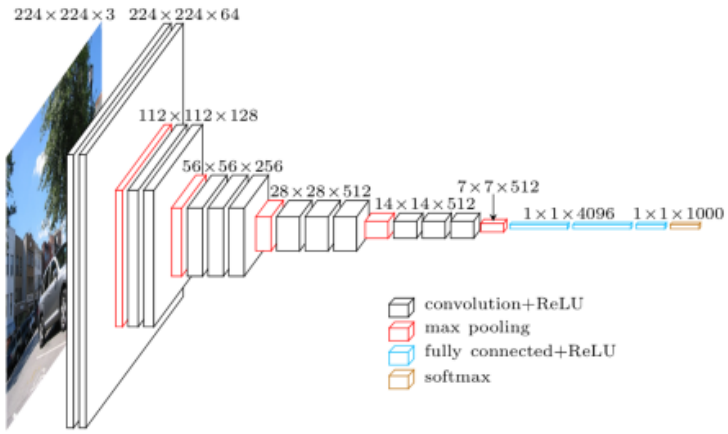


Figure 3.6: Architecture of VGG16

3.3.3 Temporal Feature Extraction

Temporal feature relates to any feature that changes over time. As we are using the video data for our work we can assume the changes of action sequentially in the videos. Therefore, we need a model that deals with the temporal feature extraction. RNNs can handle sequential data processing. Additionally, when working with the present data, RNNs are able to pick up some knowledge from the past. With extensive modeling capabilities for long-term dependencies, the LSTM and GRU are upgraded RNN models. GRU is a less complex version of LSTM. On the other hand, the bi-directional GRU have the capacity to learn information from both past and upcoming data. And, sequential data can be learned more deeply by this kind of model. So, we incorporated the bi-directional GRU in building our model. And we have considered 256 units in bi-GRU layer. After getting the temporal feature it will be passed down to the dense layer of 512 neurons. And then finally, the softmax function will predict the desired class. Below more detail is given on bi-directional GRU.

3.3.3.1 Bi-directional Gated Recurrent Unit(Bi-GRU)

GRUs and Long Short Term Memory are quite similar (LSTM). But unlike LSTM it has only two gates, they are reset gate and update gate. The update gate specifies how much of the previous information must be transmitted into the future. The reset gate specifies how much of the past information should be forgotten. It is comparable to how the input gate and forget gate work together

in an LSTM recurrent unit. The basic structure of a GRU unit is illustrated in figure 3.7. The two GRUs' states, which are unidirectional in opposition to one another, are used to construct the bi-GRU model. The first GRU goes forward, starting at the beginning of the data series, and the second GRU moves backward, starting at the conclusion of the data sequence. This enables information from the past as well as the future to affect the conditions of the present. The structure of bi-directional GRU is given in figure 3.8.

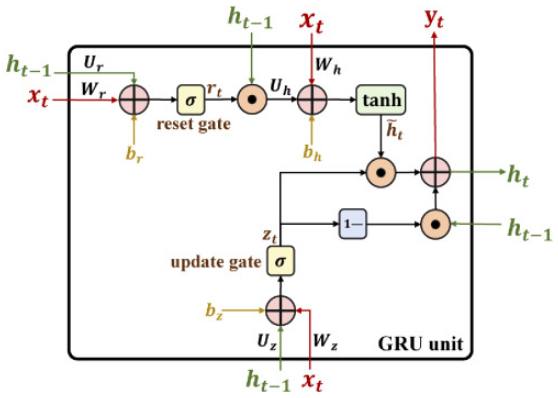


Figure 3.7: Detailed structure of a GRU unit

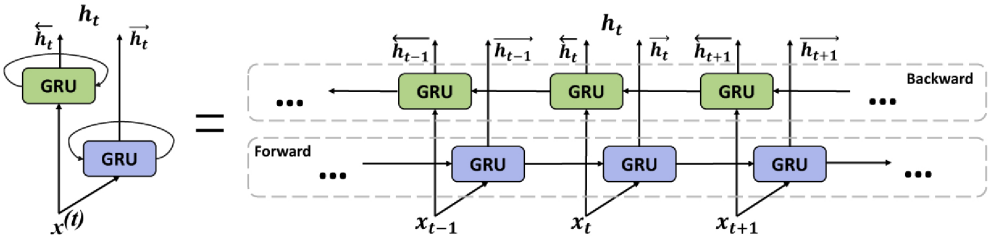


Figure 3.8: Structure of a bi-directional GRU model

3.4 Conclusion

In this chapter, we have gone into extensive detail about the methodology of our proposed deep learning-based approach to identify fall event. The experiments that we have conducted and the result of our proposed model is discussed in the next chapter.

Chapter 4

Results and Discussions

4.1 Introduction

The experimental data from our proposed approach to categorize indoor human fall incidents is analyzed in this chapter . First, we have described evaluation criteria in this chapter. Following that, we presented some experimental results related to key frame selection, target segmentation, selection of spatial and temporal feature extractor. The final discussion in this chapter compares our proposed model to previously published research.

4.2 Implementation Tools

To implement our experiments we incorporated the Google Colaboratory. Google Colab uses Python 3 with Google Compute Engine as backend. We used the free subscription which provides 12.68 GB of RAM and 107.72 GB of disk storage. We used the following libraries for our work:

- Python 3
- Keras
- Scikit-learn
- Numpy
- Matplotlib
- OpenCV

4.3 Evaluation Metrics

Here, we go over a few of the evaluation metrics that were applied to our model. The quality of a model's performance is measured by the evaluation metrics. Confusion matrix, accuracy, precision, recall, and F1-score are the commonly utilized metrics to assess the performance of a model.

4.3.1 Confusion Matrix

The confusion matrix is frequently used to evaluate how well a classification model is performing. The confusion matrix of a binary classifier will be of the following figure 4.1. The rows of the matrix contain the actual classes, whereas its columns contain the predicted classes. So, we can have four cases such as:

- True Positive (TP): The number of correctly identified falls.
- True Negative (TN): The number of accurately identified regular activities or non fall event.
- False Positive (FP): The number of regular actions that were mistakenly identified as fall event.
- False Negative (FN): The number of falls that were mistakenly identified as regular activities or non fall event.

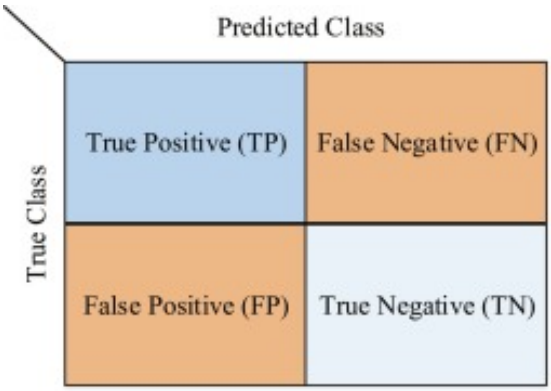


Figure 4.1: Confusion matrix

4.3.2 Accuracy

The performance of a model across all classes is summarized by the measurement of accuracy. It is beneficial when each class has equal importance. It is calculated as the ratio between the number of correct predictions to the total number of predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

4.3.3 Precision

The precision is determined by dividing the total number of correctly identified Positive samples by the number of Positive samples overall (either correctly or incorrectly). The precision measures how accurately a sample is classified as positive by the model.

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

4.3.4 Sensitivity

The recall is calculated by dividing the overall positive sample count by the proportion of positive samples that were correctly classified as positive. Recall measures how accurately a model can forecast a class.

$$Sensitivity = \frac{TP}{TP + FN} \tag{4.3}$$

4.3.5 F1-score

The harmonic mean of precision and recall is defined as the F1 score. It's a statistical metric for evaluating performance.

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{4.4}$$

4.3.6 Specificity

The ratio of actual negatives to all other negatives in the data is known as specificity.

$$Specificity = \frac{TN}{TN + FP} \tag{4.5}$$

4.4 Number of Key Frame Selection

We have conducted an experiment with different number of frames. In the figure 4.2 the accuracy rate vs the number of frames is illustrated. The accuracy of the model is seen to be increasing when the number of frames is greater than 10. Again, it starts to decrease if the number of frames is greater than 16. From the figure we can see that when the number of frames is at 16 the accuracy becomes the highest. Therefore, we chose 16 frames for our model to be processed further.

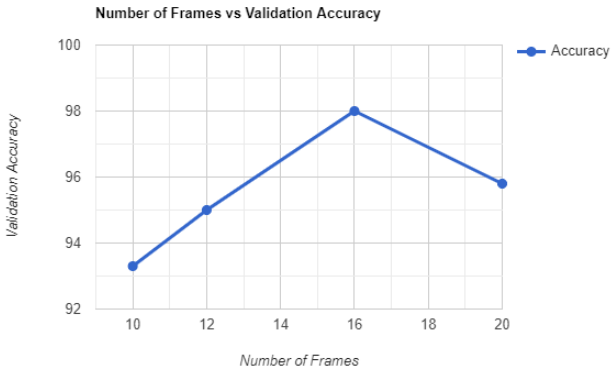


Figure 4.2: Number of frames vs validation accuracy curve

4.5 Frame Size Selection

Selecting the optimum frame size is another important task in building the model. To choose the frame size for our model, we experimented with three different sizes of frames. The sizes are 128, 150, and 224. In figure 4.3 we can observe that the frame size off 224 gives the optimum accuracy than others. Larger than 224 size will increase the computational cost as well as training time.

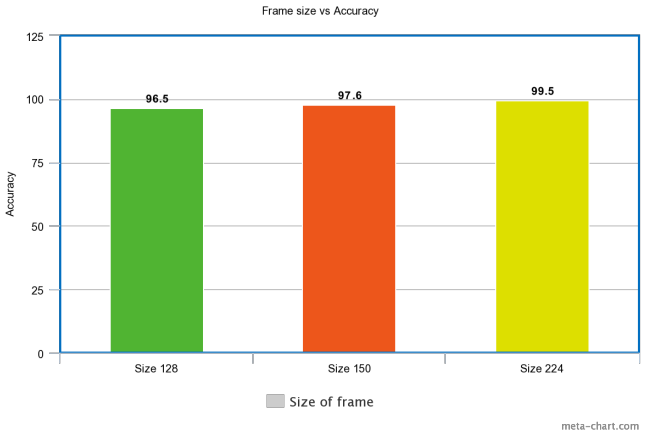


Figure 4.3: Frame size vs accuracy

4.6 Segmentation with Mask R-CNN

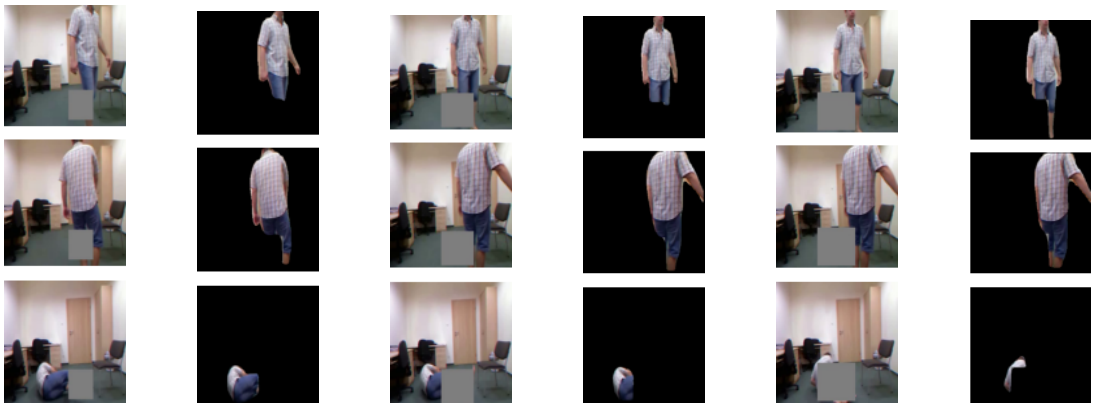
To handle the occlusion issue from our video data we incorporated the Mask R-CNN model. By locating and segmenting the occluded person our model will be able to further extract the necessary feature even with the occlusion. In figure 4.4 we can see that a fallen person is partially occluded by a chair. And with occlusion spatial feature extractor might generate unnecessary features that will affect the entire classification process. After applying the mask R-CNN,it not only segmented our target from the chair,but also subtracted the background which will decrease our computational cost as well. In figure 4.5 we can see that mask R-CNN can properly locate the target and segment it even with the occlusion. Hence, it handles the occlusion situation.

Figure 4.4: Before applying Mask R-CNN

Figure 4.5: After applying Mask R-CNN

4.6.1 Artificial Occlusion Generation

As we don't have any occlusion data in U-R fall dataset, to verify that the Mask-RCNN can properly segment our target person, we artificially generated occlusion of 20%, 50%, and 80%. In figure 4.6f we can observe the artificial occlusion that we generated for U-R fall dataset. On the other hand, as we have very few occlusion data in multiple cameras fall dataset, we also generated some artificial occlusion for evaluation. In figure 4.7f the 20%, 50%, and 80% artificial occlusion for Multiple Cameras Fall dataset is illustrated.



(a) With 20% occlusion before segmentation

(b) With 20% occlusion after segmentation

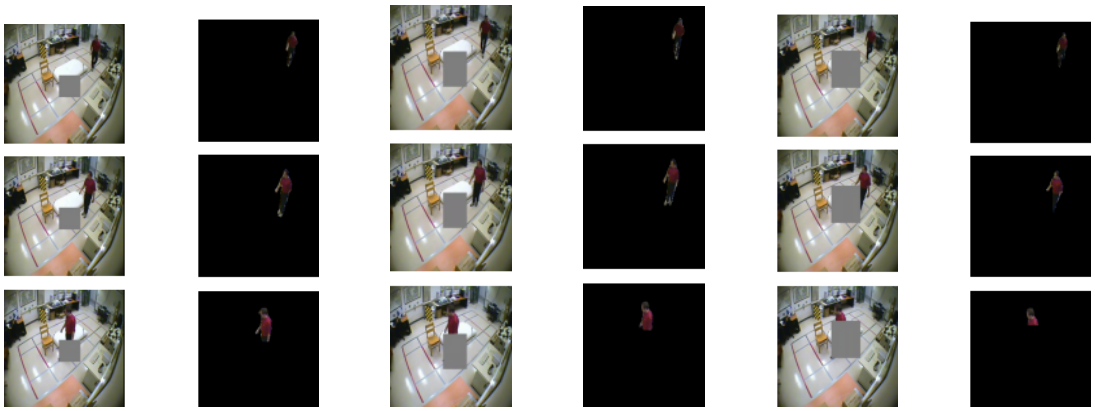
(c) With 50% occlusion Before segmentation

(d) With 50% occlusion after segmentation

(e) With 80% occlusion before segmentation

(f) With 80% occlusion after segmentation

Figure 4.6: Artificial occlusion of 20%, 50%, and 80% for U-R fall dataset



(a) With 20% occlusion before segmentation

(b) With 20% occlusion after segmentation

(c) With 50% occlusion Before segmentation

(d) With 50% occlusion after segmentation

(e) With 80% occlusion before segmentation

(f) With 80% occlusion after segmentation

Figure 4.7: Artificial occlusion of 20%, 50%, and 80% for Multiple Cameras Fall dataset

4.7 Performance of Different Pre-trained Feature Extractor

We have experimented with five different popular pre-trained models for extracting the spatial fetaures from the frames. For each of the model, the train-test-split ratio was 60:40. The input size for each model was 224×224×3. In figure 4.8 we can see the accuracy results of the five different pre-trained model that are Xception, VGG19, VGG16, EfficientNetB2, and ResNetV2. From the figure we can clearly see that, the VGG16 offers the best result with an accuracy of 99.5%. On the other hand, from figure 4.9 we can see that VGG16 comes with the minimum loss among others. Hence, we picked VGG16 pre-trained feature extractor for building our own model.

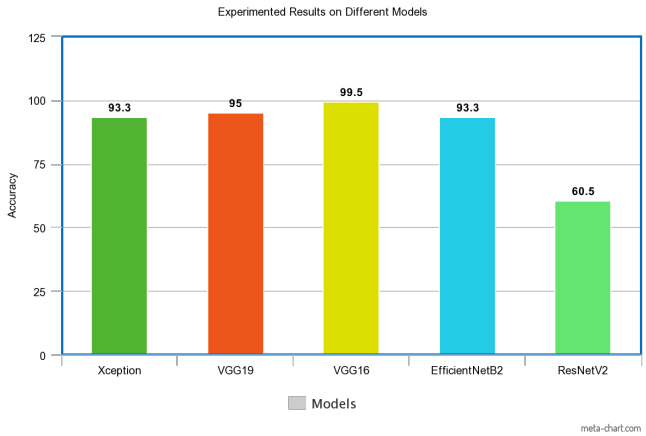


Figure 4.8: Accuracy result of different feature extractors

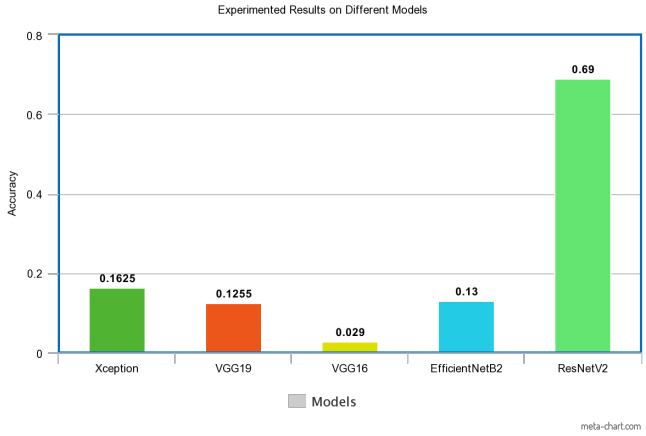


Figure 4.9: Loss result of different feature extractors

4.8 Performance of Different Sequence Models

As we are dealing with time series data, it's important to extract the temporal features. And for the task, we experimented with four different variants of RNN's. They are Long Short-Term Memory (LSTM), Bi-directional LSTM (Bi-LSTM), Gated Recurrent Unit (GRU), and Bi-directional GRU (Bi-GRU). We kept the number of units fixed at 256 for all the models. The performance of these models is illustrated in figure 4.10. As bi-GRU is giving the best performance also it takes training time less than a minute, therefore, we chose this model.

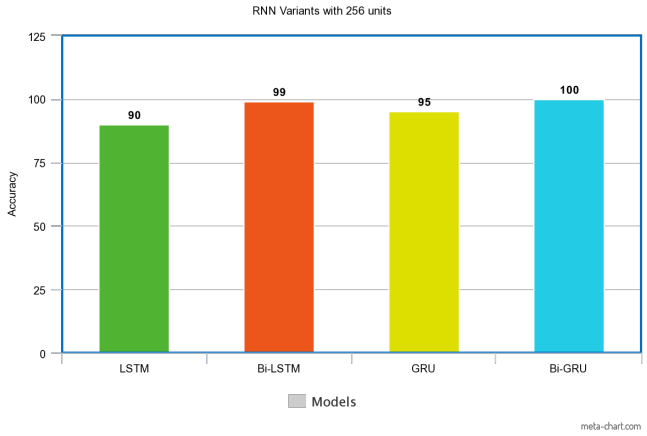


Figure 4.10: Accuracy result of different sequence model

4.9 Confusion Matrix of Proposed Model

The confusion matrix of our experiment is shown in figure 4.11 and 4.12 for UR fall detection dataset and multiple cameras fall dataset respectively. The values of blue diagonal demonstrate the number of correctly classified fall and non-fall event. From the confusion matrix of UR fall detection dataset in figure 4.11, we can see that there are total 20 test cases, and all are correctly classified. On the other hand, from the confusion matrix of multiple cameras fall dataset in figure 4.12 among 45 test cases, only one fall event is misclassified as non-fall event.

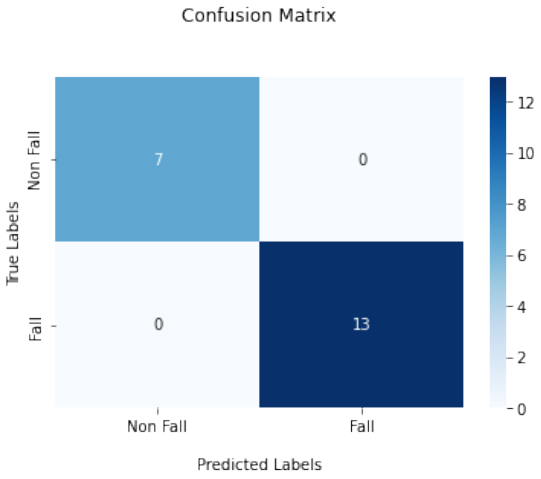


Figure 4.11: Confusion matrix of UR fall detection dataset

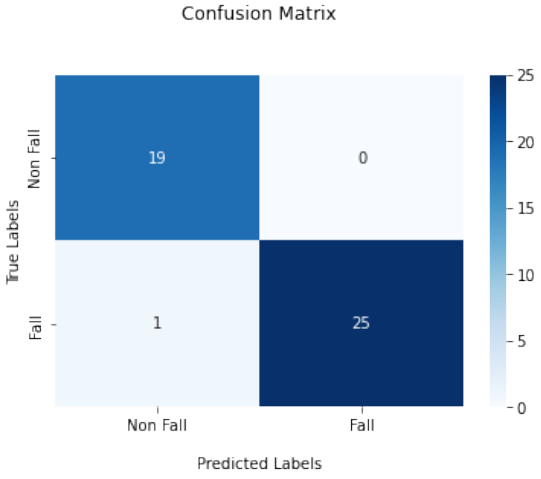
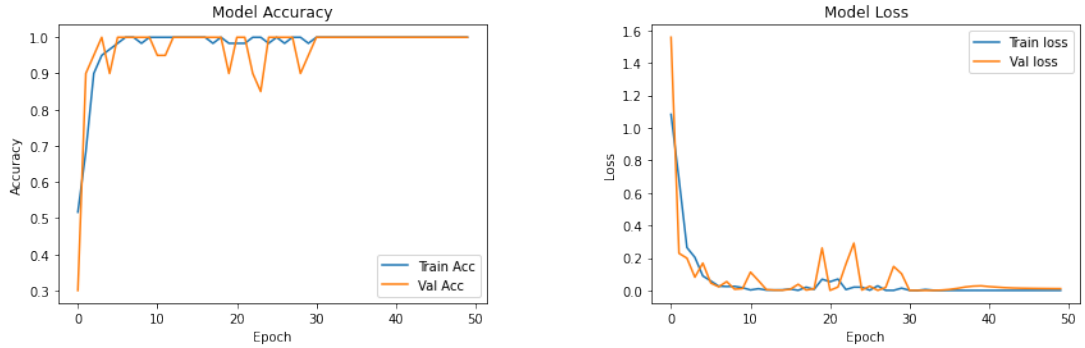


Figure 4.12: Confusion matrix of multiple cameras fall dataset

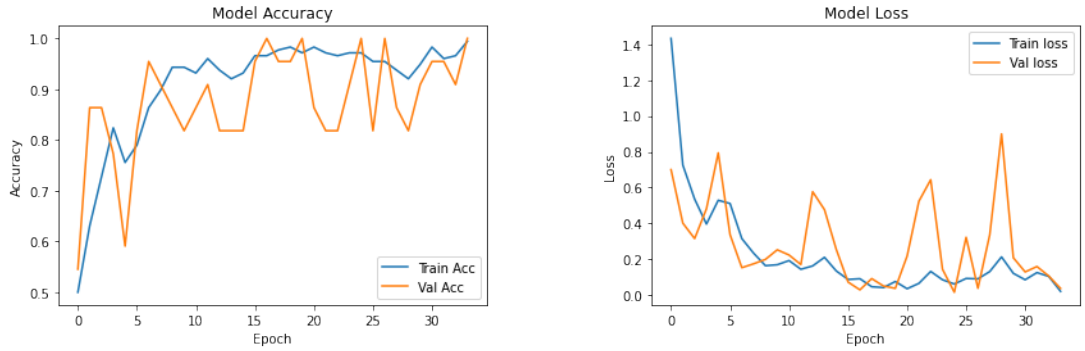
4.10 Performance Evaluation of Proposed Model

We picked the pre-trained VGG16 model for extracting the spatial features and bi-directional GRU for extracting the temporal feature. With a batch size of 16 and 50 epochs along with 60:40 ratio of train-test-split we got accuracy of 100% for UR fall detection dataset and 98.3% for multiple cameras fall dataset. We have incorporated the categorical cross entropy as the loss function and Adam optimizer with a learning rate of 0.001. However, for multiple cameras fall dataset, as we are applying the early stopping method, the model gave 98.4% accuracy at 30 epoch. The plot for accuracy and loss for UR fall detection dataset and multiple cameras fall dataset is illustrated in figure 4.13b and figure 4.14b respectively.



(a) (b)

Figure 4.13: Accuracy and loss curve of proposed model for UR fall dataset



(a) (b)

Figure 4.14: Accuracy and loss curve of proposed model for multiple cameras fall dataset

4.10.1 Performance Evaluation with Artificially Occluded Data

To verify that, our model can properly segment partially occluded data and can detect fall or non-fall event we have made artificially occluded data. We have created videos of 20%, 50%, and 80% of artificial occlusion. Total 100 videos were made with artificial occlusion. We have evaluated in all occlusion portion videos separately as well as in combination. With 20%, 50% and 80% occlusion data our model trained with U-R fall dataset gave an accuracy of 97%, 95% and 90% respectively. And with combination of all occluded data it gave an accuracy of 92%.

On the other hand, with 20%, 50% and 80% partially occlusion data our model trained with Multiple Cameras fall dataset gave an accuracy of 94%, 89% and 85% respectively. And with combination of all occluded data it gave an accuracy of 88%. The results evaluated with our artificially built occluded data in terms

of accuracy is given in table 4.1.

Table 4.1: Evaluation result (accuracy %) with different occlusion scenario

Model trained with dataset	20% occlusion	50% occlusion	80% occlusion	All occluded data
U-R fall dataset	97%	95%	90%	92%
Multiple cameras fall dataset	94%	89%	85%	88%

4.10.2 Performance Evaluation with Self-built Outdoor Dataset

As most of the dataset related to fall event detection are built within indoor environment we were not able to evaluate our model with outdoor natural light. So, in order to verify the generalization of our model, we decided to build dataset for outdoor natural lighting condition. We made total 70 outdoor videos where both single and multiple person were present. However, as we are focusing on individual person only we evaluated our model with single person data. And our model trained with U-R fall dataset gave an accuracy of 90% for our self-built single person dataset. On the other hand, our model trained with Multiple Cameras fall dataset dataset gave an evaluation accuracy of 85%.

Moreover, in order to generalize our model, we also collected data from Youtube and built another outdoor dataset. And our model trained with U-R fall dataset gave 84% accuracy. Further, our model trained with Multiple Cameras fall dataset dataset gave an evaluation accuracy of 81%. The example of our self-built outdoor dataset as well as the Youtube dataset is illustrated in figure 4.16.

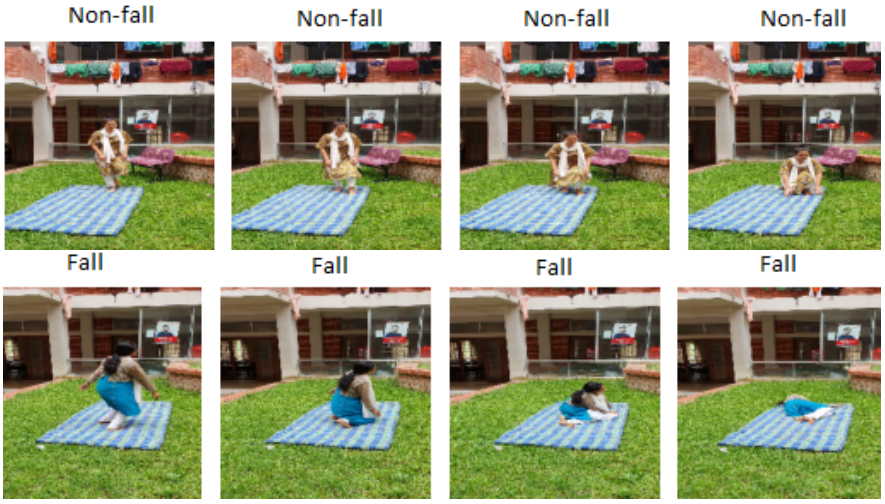


Figure 4.15: Example of self-built outdoor data

Figure 4.16: Example of Youtube outdoor data

4.11 Comparison with Existing Works

Here we will give a comparison with other research work. Table 4.2 and 4.3 illustrates the performance of our proposed model with some existing models. With our proposed model, for the UR fall detection dataset, we get the maximum accuracy of 100% which is higher than others Chen et al [16] and Lu et al [14]. Not just that, we get all the result of precision, sensitivity, and f1-score at 100%. On the other hand, for the multiple cameras fall dataset, the precision is 100% whereas accuracy, sensitivity, and f1-score is 98.4%, 96.15%, and 98.03% respectively.

Table 4.2: Performance comparison with existing methods for UR fall detection dataset

Related Papers	Accuracy	Precision	Sensitivity	F1-score
Abobkr et al [13]	98%	97%	100%	-
N. lu et al [14]	99.27%	-	-	-
Liu et al [15]	96.6%	100%	95%	-
Sultana et al [21]	99.8%	100%	100%	100%
Saha et al [23]	98.6%	96.77%	100%	-s
Our proposed method	**100%**	**100%**	**100%**	**100%**

Table 4.3: Performance comparison with existing methods for Multiple cameras fall dataset

Related Papers	Accuracy	Precision	Sensitivity	F1-score
N. lu et al [14]	99.36%	-	-	-
Sultana et al [21]	98%	100%	96%	98%
Our proposed method	**98.4%**	**100%**	**96.15%**	**98.03%**

4.12 Conclusion

The main focus of this work is to build a binary classifier model that can predict a fall or non-fall event. And to fulfill the task various deep learning models have been experimented in our work. We have seen the results of each those models with the available datasets. From this chapter we can learn that our proposed model has outperformed some existing models with a high accuracy rate for UR fall detection dataset. Not just that, our model can even deal with the partially occlusion data, which is one of our challenges. And very few works have been done focusing on this particular challenge. Also, as we considered the bi-directional GRU for temporal feature extraction it takes very less time in training. Our thesis work is summarized in the following chapter by leaving room for some possible future work that we will hopefully able to solve.

Chapter 5

Conclusion

5.1 Conclusion

Nowadays, most of the applications are utilizing vision-based devices. Infrared sensors, RGB cameras, and depth sensors are just a few of the visual sensors that have been used in fall detection activities. Since surveillance systems have become mainstream in our daily lives, RGB cameras are the most affordable and simple to install of all of them. However, several works have been proposed incorporating vision-based dataset. Therefore, we focused on improving the existing methods that comes with limitations.

We have proposed an indoor vision-based deep learning model to classify the fall and non-fall events. Because deep learning models are excellent in extracting important features and they also outperforms other existing models. As we are dealing with indoor environment it's very likely to face the challenges like illumination, occlusion, multiple person etc. And sometimes the integrated surveillance camera may also generate distorted and blurry frames that can affect the models performance. Despite all the issues, vision-based deep learning models has shown great advancement.

We incorporated the popular pre-trained VGG16 model and bi-directional GRU for building our binary classification model. And we experimented with two of the benchmark datasets that are the UR fall detection dataset and the multiple cameras fall dataset. We also chose our key frames pragmatically. The key contribution of our work is handling the occlusion incident and giving a 100% result on accuracy, precision, sensitivity, and f1-score for UR fall detection dataset. Our model has also given promising result for multiple cameras fall detection dataset with an accuracy of 98.4%.

As there is no available outdoor dataset for fall detection, we made self-built outdoor dataset for the experiment. It's been found that, even in natural lighting condition, our model gives great performance result which is 90% and 85% of accuracy for models trained on U-R fall and Multiple Cameras fall dataset respectively.

Further, for the better generalization of our model, we evaluated with the dataset that we collected from Youtube. And our approach gave an accuracy of 84% and 81% for the models trained on U-R fall dataset and Multiple Cameras fall dataset respectively.

5.2 Future Work

Comparing the current study to earlier ones, the performance is much improved. Despite that, there are some limitations yet to be solved. As the datasets are experimented in indoor simulated ideal environment, we can't know for sure how our model will handle in real-world complex environment. Also, we only used a small dataset for our experiment due to public lack of availability. In order to increase the acceptability of this research work, several potential future works are described below.

- Our proposed work has been experimented with only two benchmark datasets. In future we will try to experiment with other available datasets for building a more generalized model.
- As there was a lacking of dataset including multiple person scenario, we were not able to experiment with such case. Therefore, in future we will try to come up with a model that can handle multiple person case.
- In future we will try to build a model that will properly able to differentiate between actual fall and fall mimicking events.

References

[1] G. Sannino, I. De Falco and G. De Pietro, 'A supervised approach to automatically extract a set of rules to support fall detection in an mhealth system,' *Applied Soft Computing*, vol. 34, pp. 205–216, 2015 (cit. on p. 1).

[2] 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 (cit. on p. 1).

[3] C. Lindmeier and A. Brunier, 'Who: Number of people over 60 years set to double by 2050; major societal changes required,' *World Health Organization*, 2015 (cit. on p. 1).

[4] C. Rougier, J. Meunier, A. St-Arnaud and J. Rousseau, 'Robust video surveillance for fall detection based on human shape deformation,' *IEEE Transactions on circuits and systems for video Technology*, vol. 21, no. 5, pp. 611–622, 2011 (cit. on p. 1).

[5] S. R. Lord, S. T. Smith and J. C. Menant, 'Vision and falls in older people: Risk factors and intervention strategies,' *Clinics in geriatric medicine*, vol. 26, no. 4, pp. 569–581, 2010 (cit. on p. 1).

[6] S. C. C. Fabrıcio, R. A. P. Rodrigues and M. L. d. Costa Junior, 'Falls among older adults seen at a são paulo state public hospital: Causes and consequences,' *Revista de saude publica*, vol. 38, pp. 93–99, 2004 (cit. on p. 1).

[7] A. R. Inturi, V. Manikandan and V. Garrapally, 'A novel vision-based fall detection scheme using keypoints of human skeleton with long short-term memory network,' *Arabian Journal for Science and Engineering*, pp. 1–13, 2022 (cit. on p. 2).

[8] 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. 161 337–161 348, 2020 (cit. on pp. 2, 12).

[9] K. Sahu and S. Verma, 'Key frame extraction from video sequence: A survey,' *International Research Journal of Engineering and Technology (IRJET)*, vol. 4, no. 05, 2017 (cit. on p. 7).

[10] T. Gao, B. Packer and D. Koller, 'A segmentation-aware object detection model with occlusion handling,' in *CVPR 2011*, IEEE Computer Society, 2011, pp. 1361–1368 (cit. on p. 8).

[11] M. Roxas, T. Hori, T. Fukiage, Y. Okamoto and T. Oishi, 'Occlusion handling using semantic segmentation and visibility-based rendering for mixed

reality,' in *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*, 2018, pp. 1–8 (cit. on p. 8).

[12] K. Adhikari, H. Bouchachia and H. Nait-Charif, 'Activity recognition for indoor fall detection using convolutional neural network,' in *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*, IEEE, 2017, pp. 81–84 (cit. on p. 11).

[13] A. Abobakr, M. Hossny, H. Abdelkader and S. Nahavandi, 'Rgb-d fall detection via deep residual convolutional lstm networks,' in *2018 Digital Image Computing: Techniques and Applications (DICTA)*, IEEE, 2018, pp. 1–7 (cit. on pp. 11, 36).

[14] N. Lu, Y. Wu, L. Feng and J. Song, 'Deep learning for fall detection: Three-dimensional cnn combined with lstm on video kinematic data,' *IEEE journal of biomedical and health informatics*, vol. 23, no. 1, pp. 314–323, 2018 (cit. on pp. 12, 35, 36).

[15] K. Han, Q. Yang and Z. Huang, 'A two-stage fall recognition algorithm based on human posture features,' *Sensors*, vol. 20, no. 23, p. 6966, 2020 (cit. on pp. 12, 36).

[16] 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, 2020 (cit. on pp. 12, 35).

[17] X. Cai, X. Liu, M. An and G. Han, 'Vision-based fall detection using dense block with multi-channel convolutional fusion strategy,' *IEEE Access*, vol. 9, pp. 18 318–18 325, 2021 (cit. on p. 12).

[18] Q. Feng, C. Gao, L. Wang, Y. Zhao, T. Song and Q. Li, 'Spatio-temporal fall event detection in complex scenes using attention guided lstm,' *Pattern Recognition Letters*, vol. 130, pp. 242–249, 2020 (cit. on p. 13).

[19] W. Min, H. Cui, H. Rao, Z. Li and L. Yao, 'Detection of human falls on furniture using scene analysis based on deep learning and activity characteristics,' *IEEE Access*, vol. 6, pp. 9324–9335, 2018 (cit. on p. 13).

[20] M. Salimi, J. J. Machado and J. M. R. Tavares, 'Using deep neural networks for human fall detection based on pose estimation,' *Sensors*, vol. 22, no. 12, p. 4544, 2022 (cit. on p. 13).

[21] A. Sultana, K. Deb, P. K. Dhar and T. Koshiba, 'Classification of indoor human fall events using deep learning,' *Entropy*, vol. 23, no. 3, p. 328, 2021 (cit. on pp. 13, 36).

[22] G. Anitha and S. B. Priya, 'Vision based real time monitoring system for elderly fall event detection using deep learning.,' *Comput. Syst. Sci. Eng.*, vol. 42, no. 1, pp. 87–103, 2022 (cit. on p. 13).

[23] A. De, A. Saha, P. Kumar and G. Pal, 'Fall detection approach based on combined two-channel body activity classification for innovative indoor

environment,' *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–12, 2022 (cit. on pp. 14, 36).