Automated Human Fall Recognition from Visual Data

SUAD GOUMA ALBAWENDI

A thesis submitted in partial fulfilment of the requirements of Nottingham Trent University for the degree of

Doctor of Philosophy

February 2019

This thesis is dedicated to my mother, husband and family with great gratitude. Undoubtedly, without their prayers and support this thesis would have been impossible.

Acknowledgements

The work described in this thesis is carried out at Nottingham Trent University between November 2014 and February 2019, while I was working as a full-time doctoral research student.

I owe my utmost sincere gratitude to my honourable director of studies, Prof. Ahmad Lotfi, for his guidance over the last four years. His heart and his office always open for me at any time despite his many responsibilities. Also, his relentless effort, extensive suggestions and valuable comments to guide me completing my work were indispensable to the accomplishment of this research. I would like to extend my sincere thanks to Dr. Heather Powell for her valuable advice throughout the development of my thesis. Also, I would like to thank Dr. Kofi Appiah for his support and feedback throughout the course of this project work. Furthermore, I really appreciate and thank my sponsor, Ministry of Higher Education of Libya for awarding me this Scholarship and the cultural affairs attach at the Libyan embassy in London.

I would like to pay tribute to my family for their support, sympathy, and patience throughout the journey of this study, particularly, I would like to say thanks to my husband Salah and our lovely kids, Ola, Rama and Mohamed. Your joy and love have always put me up during the well-being and sadness of the last four years.

My thanks are extended to my father's soul who always encouraged me to carry on in my study, also I thank my mother, sisters, brothers, brothers in law and uncles. Their prayers and faith have inspired and encouraged me to continue with this study. Thanks also should be gone to friends and colleagues in the School of Science and Technology at Nottingham Trent University.

Suad Gouma Ramadan Albawendi

February 2019

Abstract

Falls are one of the greatest risks for the older adults living alone at home. This research presents a novel visual-based fall detection approach to support independent living for older adults in an indoor environment. The aim of the research was to investigate appropriate methods for detecting falls through analysing the motion and shape of the human body.

Several techniques for automatically detecting falls have been proposed. The existing technologies can be classified into three main groups of fall detectors, namely: ambient device-based, wearable sensor-based and computer vision-based techniques. Ambient device-based techniques use vibration or pressure sensors to capture the sound and vibration for detecting the presence and position of a person. Although these devices are inexpensive and do not disturb the user, the detection rate is rather low and many false alarms are generated. Wearable devices use different sensors such as accelerometer and gyroscopes to capture the human body movement information and detect falls. However, older adults often forget to wear them. Wearable sensors are also known to be too invasive as they require wearing and carrying various uncomfortable devices. Much work has been undertaken to investigate the use of visual-based sensors for fall detection using single, multiple, and omnidirectional cameras.

The proposed research reported in this thesis uses a single camera to detect a moving object using a background subtraction algorithm. The next step is to extract robust features which describe the change in human shape and to discriminate falls from other activities like lying and sitting. These features are based on motion, change in

the human shape feature, projection histogram features and temporal change of head position. Features extracted from the human silhouette are finally fed into various machine learning classifiers for fall detection evaluation.

The ability to distinguish a fall action depends mainly on the quality of the classifier inputs, therefore, the features of the extracted human silhouette play a key role in the effectiveness and robustness of detecting human falls. In this research, the timed Motion History Image (tMHI) method is applied for motion segmentation. In addition, the motion information was combined with other features extracted from the fitted ellipse around the human body to discriminate actual fall from other activities.

Fall detection methods can be divided into two main categories; threshold based methods and machine learning-based methods. This research presents threshold-based methods to distinguish between Activities of Daily Living (ADL) and falls. Fall events can be detected if the measured features values higher than pre-determined threshold values. Results show that falls can be distinguished from ADL with an accuracy of 99.82%, using our recording dataset. In addition, various machine learning methods were compared to evaluate their abilities to accurately detecting falls. Experimental results show efficiency and reliability of the proposed fall detection approach with high fall detection rate of 99.60% and low false alarm 2.62% tested with UR Fall Detection dataset. Additionally, A set of experiments have been conducted using our recording dataset, the results indicate that the proposed approach achieves high fall detection rate 99.94% and low false alarm 0.02%.

Publications

The following publications have been published as a direct result of this thesis:
Refereed Journal Papers

LOTFI, A., ALBAWENDI, S., POWELL, H., APPIAH, K. and LANGENSIEPEN, C., 2018. Supporting independent living for older adults; employing a visual based fall detection through analysing the motion and shape of the human body. IEEE Access. ISSN 2169-3536.

Refereed Conference Papers

ALBAWENDI, S., LOTFI, A., POWELL, H. and APPIAH, K., 2018. Video based fall detection using features of motion, shape and histogram. In:Proceedings of the 11th International Conference on PErvasive Technologies Related to Assistive Environments, PETRA '18, Corfu, Greece, 26-29 June 2018. New York: ACM, pp. 529-536. ISBN 9781450363907.

ALBAWENDI, S., APPIAH, K., POWELL, H. and LOTFI, A., 2016. Video based fall detection with enhanced motion history images. In:Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments, PETRA 2016, Corfu, Greece, June 29 - July 1, 2016. New York: ACM. ISBN 9781450343374.

ALBAWENDI, S., APPIAH, K., POWELL, H. and LOTFI, A., 2015. Overview of Behavioural Understanding System with Filtered Vision Sensor.In: Proceedings of the 2015 International Conference on Interactive Technologies and Games (iTAG 2015), Nottingham, 22-23 October 2015. Los Alamitos, CA: Institute of Electrical and Electronics Engineers, pp. 90-95. ISBN 9781467378741.

Contents

Nomenclature

Acronyms

AAL	Ambient Assisted Living
ADL	Activities of Daily Living
AMF	Approximated Median Filter
AMI	Ambient Intelligent
ANN	Artificial Neural Networks
BAGGING	Bootstrap Aggregation
CAFF	Convolutional Architecture for Fast Feature Embedding
CART	Classification and Regression Tree
CNN	Convolution Neural Networks
DERF	Differential Evolution Random Forest
DWT	Discrete Wavelet Transform
FFMS	Feature Feedback Mechanism Scheme
FNR	False Negative Rate
FPR	False Positive Rate
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
HOG	Histograms of Oriented Gradients
IE	Intelligent Environments
IR	Infra-Red
KNN	K Nearest Neighbour
LBP	Local Binary Pattern
LR	Logistic Regression
LSTM	Long Short-Term Memory
MEWMA	Multivariate Exponentially Weighted Moving Average
MGD	Motion Geometric Distribution

MHI	Motion History Image
MLPNN	Multilayer Perceptron Neural Network
NFI	Near-Field Imaging
ONS	Office for National Statistics
RBF	Radial Basis Function
RDT	Randomised Decision Tree
RF	Random Forest
PCA	Principle Component Analysis
PIR	Passive Infra-Red
RMS	Root Mean Square
RNN	Recurrent Neural Network
SCG	Scaled Conjugate Gradient
SVM	Support Vector Machine
tMHI	timed Motion History Image
UR Fall Detection	University of Rzeszow Fall Detection
WHO	World Health Organisation

Greek Symbols

τ	The current time-stamp
δ	The maximum time duration constant
ρ	Ratio of ellipse
σ	The standard deviation
θ	The orientation of the ellipse

List of Figures

List of Tables

Chapter 1

Introduction

The number of older adults (people aged 65 years and over) is rapidly increasing [45]. According to the UK Office for National Statistics (ONS), there will be a further 8.6 million people aged 65 or over in the next 50 years [3]. The existing social care and healthcare systems are deficient in meeting the care needs of older adults. Independent living supported by automated monitoring is an acceptable solution to deal with the increasing older adults population and limited budget available for their care. The current technologies such as Ambient Intelligence (AmI) can help older adults to live in their own homes longer. Intelligent homes (also referred to as Smart Homes or AmI Environments) deploy a range of sensors to monitor and track the presence of a person and detect anomalies with the Activities of Daily Living (ADL).

Falls among the older adults are a relatively common occurrence that can have dramatic health consequences. Studies have shown that $28 - 34\%$ of the older adults, have at least one fall every year [68]. Besides, falling is the second biggest cause of accidental death for more than 87% of older adults [112]. Falls are one of the leading causes of several types of injury including fractures, brain injuries, or strained muscles for older adults [103]. An injured person may be laying on the ground for a significant amount of time after a fall incident has occurred and before they receive assistance [42]. Fall detection and altering systems are unable to prevent falls; nevertheless, these systems can reduce complications by ensuring that falling person receive help quickly [42]. Therefore, significant attention has been devoted to develop automated human fall detection systems [125].

Research on video surveillance systems for understanding human behaviour has grown steadily in recent years [112]. Computer vision systems offer a new promising solution which can help older people stay at their own home by providing a secure environment and improve their quality of life [22]. One application area of video surveillance is to analyse human behaviour and detect unusual behaviour.

A two-way audio camera monitoring system for older adults allows to listen and view their activities. Although, activities monitoring by cameras are regarded as an intrusion of privacy and it is not acceptable by many older adults. Preferably, unobtrusive sensors should be used, along with a small computerised receiver to collect data that are then analysed and posted to a secure server for viewing by the carer/relative. The adult children/informal carers of frail older adults living alone would prefer to receive short daily report or alerts in the form of e-mail or phone calls [125].

Fall is one of the most serious causes of fatal injuries among the older adult population [103]. Most of the fall detection systems fail to distinguish between daily activities and a real fall. Therefore, this research aims to investigate appropriate approaches to detect falls through analysing the motion and shape of the human body. The research will be focused on automatically detecting human falls by using visual camera monitoring older adults in a home environment.

The main research question addressed in this thesis is to investigate the use of appropriate methods to monitor human activities in smart environments and to automatically detect falls. In particular, this study trying to answer the following questions:

- Can we detect human falls from low level camera data by using simple image processing techniques?
- Can simple detection techniques be utilised to detect fall events with fewer false detection rates?
- Can we identify falls within the data that is collected from RGB camera sensors in a real home environment?
- Can we validate and test the proposed techniques on data collected from

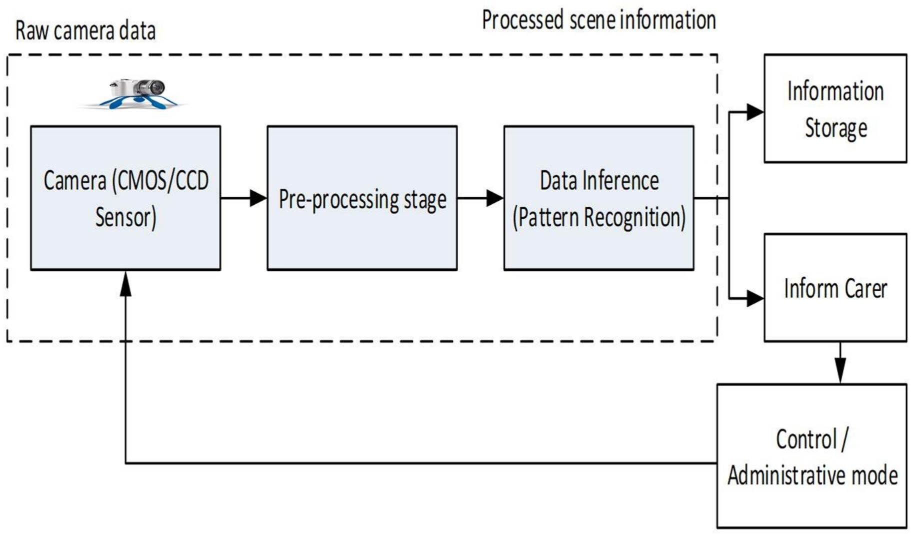

Figure 1.1: Schematic diagram of the proposed vision-based monitoring system.

real environments?

The rest of this chapter is structured in the following order. The next section presents an overview of this research. Section 1.2, aims and objectives of this study are explained. Section 1.3 introduces the significant contribution of the thesis. Finally, the remaining chapters of this thesis are outlined in Section 1.4.

1.1 Overview of the Research

This study aims to detect fall events in indoor environments based on video data. To achieve the project aim, a video-based system for fall detection is proposed. The proposed system consists of three steps; detection moving objects in a frame, tracking such objects from frame to frame and then analysis of object tracked to recognise the behaviour. A schematic diagram of the proposed vision-based older adults behaviour monitoring system is illustrated in Figure 1.1.

The pre-processing and pattern recognition stages are conducted locally and the visual information about a subject is not communicated for normal activities. The two-way audio camera communication will be permitted once an anomaly/fall is detected. This information will be shared with the carers and they will have the privilege to establish the visual communication.

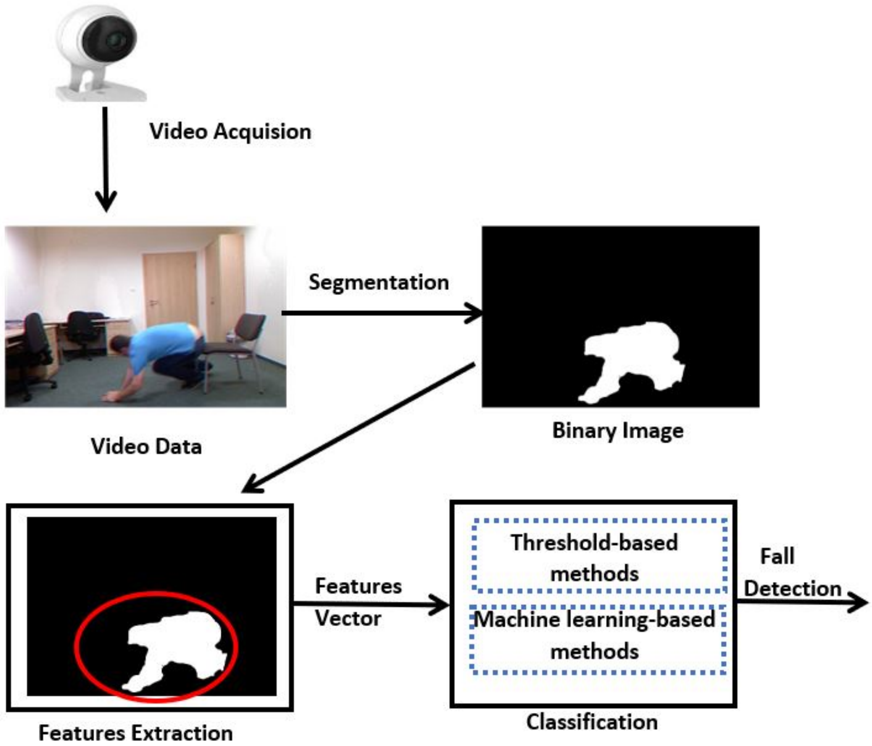

Figure 1.2: Flow diagram of the proposed human fall detection system.

This research proposes a visual-based approach for fall detection in a home environment and detecting a fall event based on motion information and the changes in human shape. An overview of the proposed fall detection system is shown in Figure 1.2. The proposed fall detection system includes four steps; data collection, foreground segmentation, feature extraction and fall detection. Background subtraction is implemented to segment out moving objects [94]. Afterwards, useful features such as motion information, a shape orientation, a temporal change of the head and histograms for detecting fall from different daily activities are extracted. The proposed method exploits motion and shape features based on the observation that human falls often involve drastic shape changes and abrupt motions as compared to other activities.

The first step in the proposed system is to analyse the motion occurring in a given time window. It is assumed that human fall has greater acceleration than other daily activates. However, focusing only on a fast acceleration can result in many false alarms during fall-like activities such as sitting down quickly [61]. Therefore, the second step is to analyse the change of the human shape to identify a fall among other activities.The analysis of the moving object is performed by fitting an approximate ellipse around the human body.

Common fall detection algorithms are based on the fact that the fall events has high motion than every day life activities. This study highlights the situation when a human fall can occur with low motion rate. A typical example is when a person loses balance and hold onto furniture to prevent a fall and yet falls on the ground. Therefore, the third feature which is projection histogram feature is also utilised to confirm a fall event. Furthermore, the proposed fall detection approach applies other features; tracking of the human head in subsequent frames to deal with occlusion problem. Finally, the extracted features are used as input vectors to various machine learning classifiers for fall and non-fall events classification.

1.2 Aims and Objectives

This research will be focused on automatically detecting human falls from camera sensor data. The work is based on a low cost, off-the-shelf visual camera monitoring system for the older adults living independently in a home environment. The aim of the research is to investigate appropriate methods for recognising and classifying falls of an occupant in a domestic room setting. In this research, a method for monitoring human activities in a home environment and detecting a fall event based on motion information and changes in the orientation regarding the shape of a person is presented.

To achieve this aim the following objectives are identified.

1. To conduct extensive research into the literature concerning existing technologies for monitoring the older adult and detecting fall events.

2. To implement pre-processing techniques on images of objects from a low-cost camera, resulting in outline detection and feature extraction to determine the physical shape, position and activity.

3. To set up a simulated domestic setting in order to recognise, track physical movement and position of a person using the techniques in 2.

4. To investigate and develop approaches to processing the results of 3. For the automatic classification of situations in order to recognise problem scenarios/situations.

5. To investigate the temporal relationship between a sequence of frames (low frame rate) to infer the presence and behaviour of the older adults from the pre-processed camera data.

1.3 Major Contributions of the Thesis

The main contributions of this thesis are:

- applying timed Motion History Image (tMHI) method to extract relevant features regarding the motion trail of a moving object over time,

- exploring the use of the major semi-axis and minor semi-axis of the ellipse fitting around the human silhouette as feature descriptors for fall detection purposes,

- introducing a new feature by computing the maximum values of foreground pixels in the horizontal and vertical histograms, as well as the difference between maximum values to effectively discriminate a fall among other activities,

- investigating a novel feature by determining the person head in each frame and compute the standard deviation of the absolute difference of y-coordinate of the head point. The implemented feature enhanced the performance of the proposed fall detection approach,

- identifying the best combinations of features capable of effectively detecting a fall,

- identification of optimal parameters for machine learning methods for fall detection.

1.4 Thesis Outline

This thesis consists of eight chapters that are summarised as follows:

Chapter 2: Literature Review - This chapter gives a review of the relevant literature in the field of Assisted Living (AL) technologies which are used in human activity recognition and fall detection domains. These technologies can be classified into wearable sensors-based methods, ambient sensor-based method and computer vision methods. In particular, the literature focuses on reviewing the use of threshold-based methods and machine learning methods for fall detection of older adults to support them to live independently in their own homes.
Chapter 3: Environment and Data Collection - This chapter describes the system to monitor the Activities of Daily Living (ADL) for the older adults and to detect falls. Three different datasets; University of Rzeszow Fall detection (UR Fall Detection) dataset [62], Le2i fall detection dataset [25] and video recorded dataset are also explained in detail to validate and test the proposed fall detection approach. Details of the collected video data are also presented.
Chapter 4: Selected Features and Tools for Fall Detection - This chapter discuss in detail the implementation of the selected features for fall detection. Details of different methods for extracting these features from binary human silhouettes such as timed Motion History Image (tMHI) method, fitting approximated ellipse around the human body, projection histograms and tracking human head in consequent frames are presented. In this chapter, machine learning methods to classify fall/non-fall activities are presented. These methods include MLP Neural Networks (MLPNN), Support Vector Machine (SVM), K Nearest Neighbour(KNN) and Bootstrap Aggregation (Bagging).
Chapter 5: Fall Detection Approach Using Threshold- Based Methods - In this chapter, the proposed fall detection approaches which employ a different combination of features like motion information, change in the orientation of the human body and horizontal histograms features are presented. These approaches are based on predefined threshold values. Experimental results of using different statistical methods which have been tested and evaluated using different datasets are presented.

Chapter 6: Fall Detection Approach Using Neural Network - This chapter provides an overview of fall detection approaches using MLP Neural Network. The experimental results of applying Neural Network methods using features mentioned in Chapter 4 are presented.
Chapter 7: Fall detection Approaches Using Various Machine Learning Methods - In this chapter, the results of applying the machine learning methods presented in Chapter 4 are validated using our recorded video datasets and UR Fall Detection dataset. A comparison of these methods is made to find the best method to detect falls.
Chapter 8: Conclusions and Future Work - This chapter provides the conclusions arisen from this thesis and formulates some future research in monitoring the daily activities of the older adults and detecting abnormal events like falls.

Chapter 2

Literature Review

2.1 Introduction

Falls are one of the leading causes of several types of injury including fractures, brain injuries, or strained muscles for older adults [103]. The latest report from the World Health Organisation (WHO) indicates that about 646,000 individuals die from falls over the world each year. Falls are the second leading cause of injury-related deaths worldwide [5]. The statistical study in [115] shows that at least one third of older adults fall one or more times a year . An injured person may be laying on the ground for a significant amount of time after a fall incident has occurred and before they receive assistance [42]. Although fall detection and altering systems are unable to prevent falls, they can reduce complications by ensuring that the falling person receive help quickly [42]. Therefore, significant attention has been devoted to develop automated human fall detection systems [125].

This chapter is structured as follows: in Section 2.2, assisted living technologies to support older people and detect fall events are introduced. Literature on fall detection methods including threshold-based methods and machine learning methods are reviewed in Sections 2.3 and 2.4 respectively. A methodology of the research is presented in Section 2.5. Conclusions are drawn in Section 2.6.

2.2 Assisted Living Technologies

An Intelligent Environment (IE) is a space in the real world which has the capability of collecting important information about an occupant's behaviour and detects gradual changes in their behaviour [73]. An Ambient Intelligent (AmI) environment can improve the lifestyle of the elderly by using different sensor technologies for example: wearable sensor-based, ambient sensor-based and computer vision-based. These technologies allow environments to be sensitive, adaptive and responsive to the presence of people in order to support them to live independently in their preferred environment [57]. Assistive technology is an umbrella term that incorporates assistive and adaptive devices for people with special needs. One important aim of assistive technology is to allow elderly people to stay as long as possible in their home without changing their living style [125].

New technologies are used to support independent living and provide security for older adults [61]. Several techniques for automatically detecting falls have been proposed [44, 70]. The existing technologies can be classified into three main groups of fall detectors, namely: wearable sensor-based [56], ambient sensor-based [131] and computer vision-based methods [91].

2.2.1 Wearable Sensor Based Technologies

Wearable devices use different sensors such as accelerometer and gyroscopes. These sensors are widely used to capture the human body movement information and generate an alarm when orientation or acceleration of the person reaches a predefined threshold [78]. Wearable sensors have several advantages in terms of low cost, power consumption and ease to use. Therefore, wearable sensors like accelerometers are used widely to detect human falls [125].

Authors in [6] propose a fall detection algorithm based on accelerometers found in mobile phones. The orientation of the mobile phone determined the orientation of the accelerometer axes. The algorithm is applies to various machine learning algorithms to a large time-series feature set to detect falls. The results show that SVM classifier can identify a fall with an accuracy of 98%.

Mao et al in [74] propose a human fall monitoring system based on a sensor unit and a mobile phone. The sensor unit consists of a tri-axial accelerometer, a tri-axial gyroscope, and a tri-axial magnetometer. The acceleration data represents the orientation of the persons body. The orientation data and Root Mean Square (RMS) of acceleration are used for fall detection. The processed data are sent into the mobile phone via Bluetooth communication. Their experiments show that the waist is the best position for sensor placement where the detection fall detection sensitivity and specificity are 100%. However, their system requires the sensor unit to be charged to ensure that the battery is adequate during collect the data and the mobile phone is connected to the sensor unit. Moreover, the sensor unit need to be fixed horizontally on the shoulder or the foot by the tape or to be placed in the coat pocket for the waist segment.

Shi et al. in [98] introduce a human fall detection system using the J48 decision tree classifier. The system employs inertial MEMS sensor which monitors the motions of the feet and waist to detect the falls. Their system achieves an overall accuracy of 98.6%, sensitivity of 98.9%, and specificity of 98.5%.

Guvensan et al. [47] developed a novel hybrid fall detection system which combines simple threshold methods with machine learning algorithms. Their system uses different types of sensors including microphones, accelerometers, and gyroscopes. The threshold value is used to discriminate between falls and normal activities. The machine learning technique is applied to classify slow fall and fall-like events, which are difficult to distinguish from actual falls. The decision tree learning algorithm of J48 is used for detection of fall events. The accuracy of their hybrid method is 93%.

Wearable sensors require older adults to wear sensor devices and some people especially those with dementia, tend to forget to wear such devices [32]. In addition, there are some wearable devices which can be activated by pushing an alarm when there is a fall. However, such an alarm can't be activated if the person is unconscious after the fall [125]. Moreover, wearable sensors are also known to be too invasive as they require wearing and carrying various uncomfortable devices especially during changing clothes and bathing [125]. Computer vision systems do not require the person concerned to wear any devices. Furthermore, such vision sensors give more information about the motion of the person, their location and their activities [43]. Moreover, the common wearable fall detectors, which are usually attached to a belt around the hip, are inadequate to be worn during the sleep and this results in the lack of ability of such detectors to monitor the critical phase of getting up from the bed [125].

One of the main reasons for non-acceptance of wearable based sensors is that the fall detectors using accelerometers generate false alarms. This means that some normal daily activities like laying or sitting are signalled as falls, which in turn leads to frustration of the users. Many attempts were undertaken to reduce the number of false alarms by combining both accelerometer and gyroscope. However, several Activity Daily Living (ADLs) like sitting down quickly have similar kinematic motion patterns to fall events and in consequence, such methods might trigger a considerable number of false alarms [32].

To solve this problem, Wang et al. [109] present fall detection system for detecting falls based on placing an accelerometer on the head level. First, a tri-axial accelerometer was placed above the ear side to measure tri-axial accelerations. Their system then computes some measurements such as the sum vector of axial accelerations and an acceleration change on the horizontal plane. A fall event is detected if these measurements reach a certain threshold value.

2.2.2 Ambient Sensor Based Technologies

Ambient sensor-based technologies use vibration or pressure sensors which are installed on the floor surface or under the bed. These sensors are used to capture the sound and vibration to detect the presence and position of a person [131]. Although, these devices are inexpensive and do not disturb the user [78], the detection rate is rather low and many false alarms are generated [37].

Many studies have also made use of sensor network systems that are able to detect abnormal behaviours of daily activity. Data may be collected from the environment using occupancy sensors such as Passive Infra-Red (PIR) movement detectors, door/ window entry point sensors and bed/sofa pressure sensors [27]. However, this technology only determines whether or not there is a moving object, and cannot extract any information about daily activities of moving object [128]. In addition, these sensors require to be installed in several rooms to cover the whole area of the monitored house [32].

For example, authors in [130] propose a system that detects human falls by using the audio signal from a microphone. Their system models each fall or noise segment by means of a Gaussian Mixture Model (GMM). Then, SVM classifier is employed to classify audio segments into falls and various types of noise. Experimental results show that the system achieves low detection rate where the accuracy of the system is 41%, precision is 70% and F-score up to 67%.

Rimminen et al. [89] present a fall detection approach using vibration sensors embedded on the floor. The approach is based on a Near-Field Imaging (NFI) which apply both floor sensor and pattern recognition to detect falls. First, the floor sensor detects patterns and location of a person and then create an image of the person touching the floor. A range of features is calculated from the cluster of observations associated with a person. These features include; the number of observations, the longest dimension, and the sum of magnitudes. After that, the Bayesian filter is used to estimate the pose of the person. Results present a 91% detection rate, sensitivity and specificity were 90.7% and 90.6% respectively.

2.2.3 Computer Vision Based Methods

Computer vision-based methods are able to give information on falls and also other daily living behaviours. Falls can be detected by employing an automatic fall detection algorithm using intelligent surveillance systems [87].

Much work has been undertaken upon the use of visual-based sensors for fall detection using single [48], multiple [122] and omni-directional [76] cameras. Recently, depth camera such as Microsoft Kinect 2.0 [18] and ASUS XtionPro Live [17] have also been used for detecting falls. The Kinect is a motion sensing technology which combines an RGB camera and a depth sensor to track the moving object in 3D [100]. The depth camera, such as Microsoft Kinect has been used for fall detection. The depth camera of the Kinect can work well in low light condition and when the light condition significantly changes such as switching on or off the light [18].

Depth cameras are used widely for human activity recognition. For example, the authors in [51] propose a novel skeleton-based approach to describe spatio-temporal aspects of a human activity using three-dimensional (3D) depth images. Two feature channels are calculated from the 3D joint position including the spatial and the temporal aspects of the activity. These features are used as inputs to Extremely Randomised Trees algorithm to train the human activity model. The experimenters are conducted using the Microsoft Research MSR 3D Action dataset [65]. The trained classifier model achieved an accuracy of 80.92%. Similarly, authors in [36] proposed a method for skeleton-based human action recognition. Motions between rigid bodies are used to describe human posture and draw movement components and mapping them to the points on a Grassmannian manifold. Then representative postures are extracted through spectral clustering. An action is represented by a symbol sequence generated with a global linear eigenfunction. Finally, the Hidden Markov model (HMM) is used to classify these action sequences. The recognition rate is over 80%.

Thermal cameras are also used to track a thermal target and analyse its motion. The work in [106] presents a novel approach for fall detection with Infrared (IR) thermal array sensors. Features including the maximal temperature difference, the maximal motion distance, the duration of motion and variance of the maximal thermal difference between foreground and background over time are extracted for fall detection. The system achieves fall recognition rate over 94% at room temperature (up to $24°C$).

Kido et al [59] present fall detection algorithm by using a thermal camera to detect falls in the toilet room. The algorithm based on the temperature difference between the person and the toilet room. Their technique achieves an accuracy of 97.8%, while the room temperature needs to be $31°C$ or less. Thermal camera can be used in temperature-managed facilities such as hospitals. However, thermal imaging technique could be unsuitable for detecting falls at home, because temperatures are not often managed consistently inside homes.

Thermal imaging applications abound in the field of human healthcare, thermal camera is being used to help detect cancer earlier, locate the source of arthritis and detect human veins. For an example; The work has been done by Asrar et al. [14] uses three types of cameras; a visual, an infrared and near infrared for the detection of veins to support the cannulation process. The collected images from three cameras were analysed using image processing techniques and compared with identification templates to evaluate the performance of each technology. The results show that the near infrared technology supported by suitable LED illumination provides the most accurate detection of veins. Where the infrared technology with the use of a cold compress helps to enhance the visualisation of veins.

Elgargni et al. [40] employ infrared and vision systems to detect the existence of the tool in the spindle and to assess whether its health whether is normal or broken. They develop a software program for tool tracking and health recognition based on Principal Component Analysis (PCA) and Discrete Wavelet Transform (DWT) combined with neural networks. Both infrared and visual cameras are used to locate and track the cutting tool during the machining process. The methodology proposed develops an improved computer algorithm for tool recognition and evaluates the performance infrared and visual data. The evaluation of the use of DWT combined with feedforward neural networks with one hidden layer is divided into two stages; the first stage evaluates the ANN based on infrared image data and DWT, the second stage evaluates the ANN based on visual image data and DWT. The results show that the use of infrared data processed using DWT and neural networks achieves a 100% success rate.

Fall detection approaches can be divided into two main categories; threshold-based approaches or a machine learning-based approaches. Threshold-based approaches detect falls by checking if the measured feature exceeds a predefined threshold value [64]. Machine learning-based approaches use labelled data to train a classifier using supervised machine learning algorithms such as Support Vector Machines (SVM), Decision Tree, and Artificial Neural Networks (ANN) to recognise the characteristic features of falls.

2.3 Threshold-Based Methods

Threshold-based methods are simple and have been applied widely in existing fall detection systems as they have a low computational cost [54]. These approaches use manually pre-defined threshold values and they do not require a learning step to classify falls [64]. However, manually defining thresholds is difficult, as several activities of daily living (ADLs) like quickly sitting or laying can produce high acceleration which can cause a high number of false alarms. In addition, the high number of false alarm is generated due to manually predefined threshold values do not generalise well for unseen persons [52]. In contrast, some falls may have a lower acceleration which leads to falls undetected [107]. This is mainly because some sensors are sensitive to the presence of noise in the living environment [124]. Several studies used machine learning algorithms as an alternative to reduce the number of false alarms while maintaining a high detection rate [86].

Mastorakis and Markis [75] introduces a novel fall detection system based on the Kinect camera. The algorithm analises the human 3D bounding box and employs thresholding on the first derivative (velocity) of width, height and depth to determine whether a particular event is a fall or not.

Rougier et al. [90] propose a fall detection method based on two features; 3D body velocity, and human centroid height relative to the ground. Firstly, a Kinect camera is used to collect depth data, and then the foreground object is segmented from a background image. Their method uses the centroid height of the person to detect falls ends on the ground. The body velocity feature is used to deal with the occlusion problem, for instance, a person is fallen and occluded behind furniture. A fall is detected when the centroid velocity of the person is above a certain threshold while the distance of the centre of mass to the floor is below another certain threshold. The experiments were run in a laboratory setting, including 54 non-fall samples and 25 fall samples. The results show that only one fall was notdetected and no false alarms were reported. Although the authors claim that their method achieves an overall detection rate of 98.7%, their method requires floor coordinates to operate and the dataset has a very limited number of samples. Additionally, some essential details about the collected dataset missing such as the number of subjects performing the experiments, the type of falls, and whether or not the data contains fall-like events such as lying or sitting quickly on a sofa.

Kong et al. [60] introduce an algorithm for fall detection by using a depth camera. Firstly, the algorithm extracts the binary image to detect human blob. The median filter removes the noise in the binary image. Then a canny filter is applied to get the outline of the binary image. After that, the algorithm computes all the white pixels in the outline image. The tangent vector angle of each white pixel is computed and divided into 15 groups. Finally, their algorithm detects falls if most tangent angles are below 45°. The experiments were collected in a living room which includes over 700 images. The accuracy of their algorithm is 97.1%, where, sensitivity and specificity are 94.9% and 100% respectively.

A novel method for fall detection was proposed by [122] through analysing dynamic shape and motion of human body regions on Riemannian manifolds. The method represents human activities by dynamic shape points and motion points moving on two simple Riemannian manifolds. Afterwards, the velocity statistics on the two manifolds are computed. The test results show a high detection rate of 99.38%.

2.4 Machine Learning-Based Methods

As an alternative to threshold-based methods, machine learning methods are widely used to classify falls. In these methods, the data sequences is usually split into a number of segments or sliding windows. There are two types of sliding windows; a fixed-length overlapping sliding window and fixed-length non-overlapping sliding window. Then, features are extracted from all data segments and fed into one or more machine learning classifier. The machine learning approaches combine both features and labelled data to train a classifier using supervised machine learning algorithms to classify fall events. This section summaries most cited machine learning methods for human activity recognition and fall detection.

2.4.1 Random Forest

Random Forest consists on an ensemble of classifiers with low bias and variance performances. RF is widely employed for solving classification and regression problems [80]. The study done by Nunes et al. [80] introduces a framework for human daily activity recognition using depth data. The first step is to divide each activity into actions of variable size identified by key poses. A pre-processing step is applied to the 3D skeleton data to normalize the data. Then, static and dynamic or temporal features are extracted from each action window. Static features represent positions of the skeleton, defined as key poses. On the other hand, dynamic features provide information about the skeleton motion, described in terms of joint movements between key poses. The static features include; projected distances between two joints and projected angles based on three joints. Dynamic features include; velocities of joints coordinate and projected angular velocities. The extracted features are used to train a Random Forest (RF) classifier. Then, an extension of the RF classifier named the Differential Evolution Random Forest (DERF) algorithm is proposed. The main advantage of the DERF algorithm is that it has no thresholds to tune, but a few adjustable parameters with well-defined behaviour. Their activity recognition framework achieves a precision of 81.83% and recall of 80.02%.

Authors in [12] propose a real time system for fall detection via computing various temporal and spatial features from the foreground object. The algorithm is based on the variances of various temporal features. These variances include temporal variations of the aspect ratio of the bounding box, the orientation and the ratio of the fitted ellipse around the foreground object, the motion vector, the upper half area of the bounding box and the geometric centre position. The aspect ratio refers to the ratio of the width to the height of the bounding box enclosing the foreground object. The orientation is computed as the angle between the major axis of the ellipse containing the foreground object and the horizontal axis of the image. The ratio between the length of the major and minor axis of the ellipse. The change of the motion is the variation in the position of the foreground object between the current and the next frame. The upper half area of bounding box was used as a shape descriptor to capture the variation in the orientation of the upper part of the body when a fall occurs. The temporal variation in the x-component and y-component of the centre of the current frame are computed to provide the temporal change in the geometric centre position of a foreground object. During a fall, the values of variation in motion and orientation are considerably high and changed rapidly when a person fell from standing state to a fall state. Features are fed into boosting with J48 and Adaptive boosting (Adaboost) classifiers. Experiments were run on publicly available datasets including the Multiple Cameras Fall dataset [15] and the UR Fall Detection dataset achieving accuracy of 99.2% and 99.0% respectively.

2.4.2 Hidden Markov Models

Hidden Markov Models are applied in many studies to model human behaviour and detect abnormality in the daily routine. For example, Dai et al. [30] present an image-based method for fall detection by applying an statistical human posture sequence model. A Kinect camera is used to extract the skeleton view of a human body in depth images. Hidden Markov Models (HMMs) are trained by the labelled extracted features to distinguish between various fall avents and daily activities. Firstly, principle component analysis is applied to reduce the data dimensionality. Then, the k-means clustering algorithm is employed to prepare training data for HMMs. The experimental results demonstrate an average fall recognition rate above 80%.

The study reported in [28] introduce an approach for temporal detection of social interactions. Their approach temporally detects intervals of individual or social activities from continuous video streams of RGB-D data. They develop a computational model for the temporal segmentation of human interactions. The model is based on features extracted from the upper body joints of the skeleton, such as: shoulder, head and torso joints. Then, the normalisation step was performed using the minimum and maximum values for each feature. Furthermore, a median filter is applied to reduce noise. Finally, these features were feed into two standard models; HMM and SVM. Their approach achieves an accuracy of 85.56%.

2.4.3 Support Vector Machine

Support Vector Machines are widely used for detecting abnormal human behaviour in data collected from intelligent environments. For instance, Kwolek and Kepski in [62] present a low cost embedded system for fall detection. The system is based on acceleration data and depth maps. A tri-axial accelerometer measures the acceleration of the person and the rate of change in velocity across time. If the measured acceleration is higher than a predetermined threshold value, that indicates a possible fall have happened. The next step is to use depth images from Kinect camera. The algorithm extracts the foreground object, calculates the features and then the feature vector is fed into the SVM classifier to make the final decision about the fall. The features extracted from depth images are; a ratio of width to height of the persons bounding box, a ratio of the height of the persons bounding box in the current frame to the physical height of the person, and the distance of the persons centroid to the floor. The experiments were conducted on the UR Fall Detetion dataset. The performance of the system using only depth images achieves an accuracy of 90%, precision 83.30%, sensitivity 100% and specificity 80%. The results show that their system using depth images with acceleration data achieves an accuracy of 98.33%, precision 96.77%, sensitivity 100% and specificity 96.67%.

Harrou et al. [50] introduce a fall detection approach based on the combination of a Multivariate Exponentially Weighted Moving Average (MEWMA) monitoring scheme and a SVM classifier. Firstly, the human body silhouette is extracted from each frame and then the body is divided into five areas. These areas are obtained by defining five lines from the silhouettes center of gravity. Experiments were conducted on UR fall detection database. The results report an accuracy of 96.66%, sensitivity of 100% and specificity of 94.93%.

Debard et al. [34] present a feature-based approach for human fall detection. The approach based on four features include; the angle of fall, aspect ratio, centre velocity and head velocity. These features were fed into the SVM classifier. Various combinations of features were tested. Their experimental results reported that fall angle, head velocity and aspect ratio is the best combination of features. Their approach achieves a recall of 89.6%. However, the system is unable to discriminate between falls and sitting down activities.

Charfi et al. [24] propose an automatic fall detection approach in a home environment. The system is based on a single camera to extract RGB video images. A number of features are exploited from a bounding box and also from a fitted ellipse around the human body. The features extracted from the bounding box are: height and width of the bounding box, the aspect ratio and the coordinates of the centre of the bounding box. The features extracted from the ellipse are: the orientation of the ellipse and the coordinates of the centre of the best fitting ellipse. In addition other features like horizontal and vertical projection histograms and moments of order 0, 1 and 2 were calculated. Several transformations including Fourier Transform, Wavelet transform, first and second derivatives are applied to the features. After testing various combinations of features, the results show that a set of seven features including; the aspect ratio of the bounding box, Y coordinate of the center of the bounding box, X coordinate of the center of the ellipse, the orientation of the ellipse, the $m00$ and $m02$ moments and the horizontal projection histograms are the best features combination. The SVM classifier is fed by the set of selected features and the system archives 98% of sensitivity and 99.60% of specificity. Experiments are done on Le2i fall detection dataset. However, it is not clear how this system would perform in other dataset as the thresholds were manually defined for the Le2i fall detection dataset.

Zhang et al. [126] introduce an automatic fall event detection method from both 3D depth and 2D RGB appearance information. Firstly, kinematic features such as vertical height of the tracked person are extracted from 3D depth images. The 2D images are used to extract two features; the ratio between the width and the height of the bounding box, and histograms of width-height ratios of the monitored person. The Kinect camera provides 20 body joints tracked for each person in each depth frame, however only 8 body joints on head and torso information are choosen to distinguish whether a person is falling. Their method aims to recognise five activities including three normal activities: standing, sit on chair, and sit on floor and two fall events including; fall from standing and fall from chair. The hierarchy SVM classifier is shown to robustly recognise the falls. Experimental results show that the method achieves an accuracy of 98%.

Bian et al. [18] propose a fall detection approach using a single depth camera. A Randomized Decision Tree (RDT) algorithm is employed to extract key joints of the human body. Then, the head joint distance trajectory are used as input feature vector to the SVM classifier to detect fall events. The video data collected in a laboratory setting containing 380 samples has equal number of fall and non-fall samples. The results demonstrate that all falls were detected and 9 false alarms were generated. However, this approach cannot detect the falls ending lying on furniture since the distance between the body and the floor is high.

Wang et al. [110] propose vision-based fall detection method using multiple cameras. First, a background subtraction algorithm is used to extract the moving person. Second, a new feature named HLC is introduced. The HLC feature is a combination of three features; **H**istograms of Oriented Gradients (HOG), **L**ocal

Binary Pattern (LBP) and **C**onvolutional Architecture for Fast Feature Embedding (Caffe). The HLC feature is used to represent a motion state of a person in a frame of a video sequence. The method includes two training stages. In the first training stage, the images are extracted from the video sequences and classified into three activities: walking, falling and lying. The SVM classifier is trained by the extracted three features from the input images. A single frame classification model is obtained from the first training stage. The single frame classification model is then used to classify every 30 continuous frames in a video sequence into fall or non-fall categories. In the second training stage, the categories of the 30 frames are used as input to the fall detection model to decide whether a person is fallen or not. Their method achieves 93.7% sensitivity and 92.0% specificity.

2.4.4 K-Nearest Neighbour

The K-Nearest Neighbour (KNN) classifier has been widely applied as an effective classification model. For example, Kwolek and Kepski in [63] introduce a fall detection system based on an accelerometer and a kinect sensor. The accelerometer wirelessly transmits the motion data to the embedded system. The Kinect sensor is used to acquire depth images in order to obtain lower false alarm ratio. The system then continuously updates the depth reference image and extract features. Finally, the system employs a KNN classifier for fall classification. Experiments show that the KNN classifier obtains good results on UR Fall Detection dataset in terms of sensitivity and specificity. The accuracy of the system was 95.83%, precision 90.91%, sensitivity 100% and specificity 92.86%.

Recently, Kwolek and Kepski [58] presented an event-driven system for fall detection based on data from a body-worn accelerometer and sequences of depth maps from a Kinect sensor. If a significant change on the persons motion is detected by the accelerometer, the system validates the fall event from depth maps. Their study evaluates different algorithms for fall detection on the basis of two main modes; the first mode uses depth maps provided by a depth sensor facing the scene, whereas, the second mode uses depth maps acquired by an active ceiling mounted camera. To deal with data maps acquired by an active camera, the algorithm tracks the centroid of the person in each frame and detects the persons head. Then, the algorithm extractes some features such as the distance between the persons head and floor and the centroid-floor distance. The fall detector system uses the features extracted from both facing and overhead cameras as input features to both a linear SVM and K-NN classifier for lying pose detection. Their system obtains an accuracy of 100%.

The work in [32] introduces a fall detection approach based on artificial vision algorithms. The first step is to use a single camera to acquire video data. The second step is to apply the background subtraction algorithm to extract the foreground subject. Then, a Kalman filter is used to reduce noisy data. The next step is to extract some features from the foreground subject for instance; the orientation of the ellipse enclosing the subject, the ratio between the width and the height of the rectangle that encloses the subject and the derivative value of the absolute normalised ratio variation. These three features describe how quickly the human silhouette changes over time to distinguish activities like walking, standing, sitting and falling. The final step is to apply the KNN classifier to classify the current activity of the extracted foreground subject. Their results show an accuracy of 96.9%, sensitivity of 96% and specificity of 97.6%.

Putra et al. [86] propose an event-triggered machine learning approach for fall detection using wearable sensors. Firstly, a fall event is divided into three stages:(pre-impact, impact, and post-impact). Then, a finite state machine is used to align each segment to one of the fall stages. Several features were used during the training and testing processes of the classifier. These features are; minimum, maximum, and average acceleration vector magnitudes, variance of the acceleration vector magnitude, Root Mean Square (RMS) of the acceleration vector magnitude and velocity. To evaluate their approach, experiments were conducted using two publicly available datasets; Cogent Dataset [83] and SisFall [104]. Finally, various machine learning methods were applied to classify fall and non-fall events. These methods include Classification and Regression Tree (CART), Logistic Regression (LR), KNN and SVM. Experimental results shown that, their approach achieves F-scores of 98% for a chest-worn sensor and 92% for a waist-worn sensor.

2.4.5 Artificial Neural Networks

Artificial Neural Networks (ANNs) have been used to distinguish between normal and abnormal human behaviour patterns from low-level sensors. For example, Harrou et al [49] proposed a statistical approach to detect human falls based on both video data and acceleration data. Video data was collected via camera and accelerometer named X-IMU inertial sensor. A Shewhart control chart is used to detect a fall by using the accelerometer data. Features are extracted from images which contain a human body silhouette. The silhouette is divided into five areas. The set of ratios that are computed for each frame are then computed to form the feature vector. The features are used as input data to Neural Network. The experiments were conducted on UR Fall Detection dataset. Their system achieves an accuracy of 96.67%, sensitivity of 100% and specificity of 93.4%.

2.4.6 Deep Learning Methods

Nowadays, the use of deep neural network is growing in many problem domains, including vision-based fall detection. The authors in [81] propose a vision-based fall detection approach using Convolution Neural Networks (CNN). The optical flow images of consecutive frames are used as input to the CNN. These images represent the motion between consecutive frames. A number of features are extracted by a modified $VGG16$ architecture to process various frames and extract motion. Initially, the CNN is trained on small fall datasets to acquire the relevant features for fall detection and then the CNN is sequentially trained on different datasets. Finally, transfer learning is used by reusing the network weights and fine-tuning the classification layers. The results show an accuracy of 95%, 100% sensitivity and 92% specificity, when tested on the UR Fall Detection dataset.

Wang et al [111] propose a fall detection framework based on automatic feature learning methods. First, the extracted frames from video images including the foreground blob are used as the training set. Then a simple deep learning model; PCANet is trained by using all samples to predict the label of every frame. The predicted labels from the trained PCANet model are then used by the SVM classifier to detect falls. After training; two models are obtained; a single frame detection model and an action model. The training process includes two stages:

in training stage 1, the training samples are labelled to three classes namely, standing, falling and fallen and then a PCANet model is trained by all samples in order to label every frame. The PCANet is used to extract the features of the images using all information from the bounding box containing the foreground blob. In training stage 2, the extracted features from PCANet with the predicted labels are used to train an SVM for fall classification and to obtain an action model. Their system achieves an 89.2% sensitivity and 90.3% specificity.

Doulamis [39] proposes a fall detection framework based on the joint estimation of a foreground object and motion information. The motion vectors on particular selected points on the image plane and the vertical velocity of the upper boundary of the foreground object are estimated to detect falls in different directions from the camera position. Recently, Doulamis [38] introduced a vision-based approach to the problem of human fall detection by using a self-adaptable deep machine learning approach. The deep learning approach is employed to detect a foreground object from the background. Then, three features are extracted including the human height, the vertical motion velocity of the object and the height-width ratio for fall detection.

The authors in [77] propose a fall detection method for falls against furniture such as sofas and chairs. Their method is based on the activity characteristics of the detected people such as motion speed and human shape aspect ratio. Convolutional NN is employed to obtain the information of the locations of the objects in the scene. The method can distinguish falls from other activities with an accuracy of 95.50%.

Yi-zeng [53] introduces an optical flow feedback convolutional neural network system for human fall detection. Firstly, the system employs a median filter to extract the silhouette in depth images. The system is based on optical flow image that are used as input to the convolutional network. The reduction algorithm is applied to obtain the shape of the person from the depth data. The next step is to extract some features such as the orientation of the ellipse and the angle between the shape of the body and the lower level. The system tests if two of these measurements are higher than threshold values, then a fall event will be recognised. A Feature Feedback Mechanism Scheme (FFMS) is proposed to allocate the features of the convolutional layers with the best object recognition models.

The FFMS uses interest of optical flow vector to calculate the Euclidean distance of histogram and use them as the input to the convolutional layer to determine the motion object boundary. During the training process, the Euclidean distance of two consequent images is calculated before feeding into the convolutional layer. Additionally, the system adopted the 3DCNN temporal information to construct the rule-based motion event. The rule-based motion is used to filter the motion event. The back-propagation algorithm was used to train the deep network. To evaluate the performance of the system, their proposed model was tested on the the KTH [93] dataset. The results show an average accuracy of 92.65%.

The work presented in [99] proposes a deep learning-based approach for human fall detection, using a Long Short-Term Memory (LSTM) neural network. The 3D locations of the body joints are used as features for discriminating different activities. As activities might take a long sequence of frames, the traditional Recurrent Neural Network (RNN) encounters the vanishing gradient issue. For this reason, the LSTM was introduced as a modified version of RNN to solve this issue, so that the neural network can handle long frame sequences. However, the LSTM requires large amount of training data, thus, transfer learning is employed in training the LSTM. In the training process, a multi-class LSTM is trained using training samples extracted from daily activities. The weights of the first few layers are copied to a two-class LSTM for fall detection. The last layer of the two-class LSTM was trained on fall samples in combination with daily activities samples. The experiments were conducted on the NTU RGB+D Action Recognition Dataset [97]. Their method achieves 93% precision and 96% recall.

The study presented in [102] introduces a video-based fall detection approach using stereo camera data. The approach starts by using a 2D human pose estimator in combination with a CNN to extract 3D human pose. Then, the ground plane in 3D was calculated. Finally, the approach uses multiple measures such as head position to detect whether a person has fallen. The results show accuracy above 91%.

Zhang et.al. [127] present an approach named Trajectory weighted Deep convolutional Rank pooling Descriptor (TDRD) for fall detection. First, a Trajectory map was used to extract a CNN feature map of each frame. Next, the CNN feature map of each frame is weighted with its corresponding trajectory attention map to get the trajectory-weighted convolutional visual feature of human region. Then, a cluster pooling method is used to reduce the redundant frames of the videos. Finally, the rank pooling method is invoked to encode the dynamic of human actions to get TDRD. Furthermore, TDRDs are used with a SVM for fall detection. Experimental results on UR Fall datasets, show that the TDRD approach gets 100% sensitivity and 95% specificity.

2.5 Methodology

This research presents a novel visual-based fall detection approach to support independent living for older adults. The proposed approach includes four steps; data collection, foreground segmentation, feature extraction and fall detection. The first step is to detect a moving object from camera data using a background subtraction algorithm [31]. The second step is to extract number of features which describe the change in human shape and allow discrimination of falls from other activities like lying and sitting. These features are based on motion, change in the human shape, projection histogram and temporal change of head position.

The first stage of the system is to analyse the motion occurring in a given time window, using tMHI [66]. Analysis of the moving object is performed by fitting an approximate ellipse around the human body [41]. After ellipse fitting, the orientation of the ellipse and the ratio between the major semi-axis and the minor semi-axis are taken as features to describe human body posture in a general way. A bounding box was used to surround the foreground object, then the y-coordinate of the top left point of the bounding box was computed and the absolute difference of y-coordinates in successive frames were used as features.

In order to evaluate the proposed human fall detection approach, video data was recorded in a realistic home environment using a single camera. Participants were selected with varied age, gender and height. The data recording was conducted in a controlled environment and the participants performed normal day to day activities such as: walking, sitting, bending, lying on the sofa and simulated falls. The experiments were carried out in different environments; bedroom, kitchen, dining room, office and living room. Additionally, the experiments were conducted using publicly available datasets; UR Fall Detection dataset [62] and

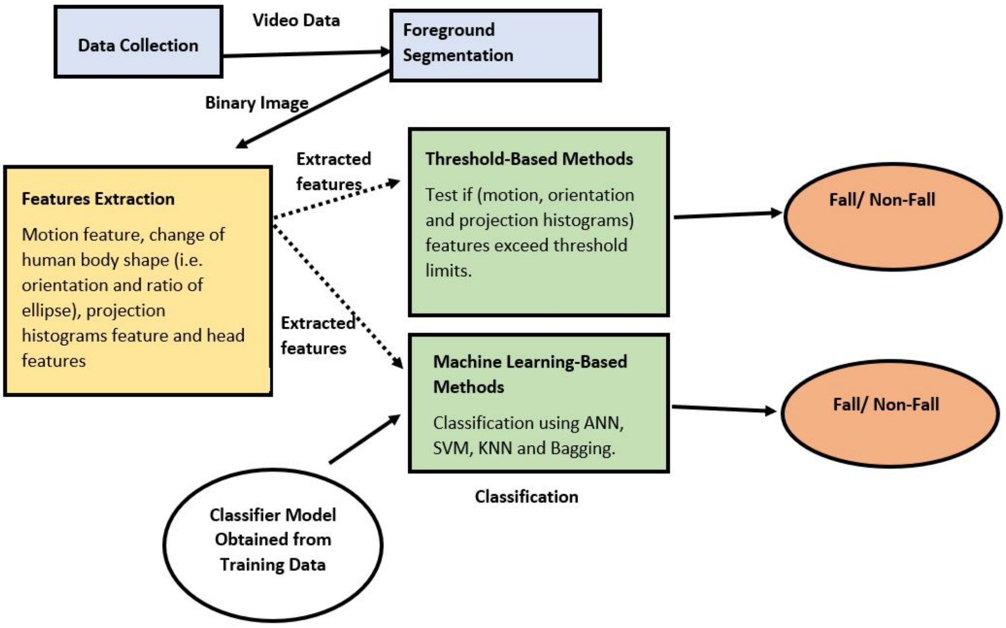

Figure 2.1: Phases of the research (overall picture).

Le2i fall detection dataset [25].

After the data acquisition process, each video data is converted into a number of frames. The acquired video data is used to extract the relevant features, providing the inputs for the fall detection methods. Fall detection methods can be divided into two main categories; threshold-based methods and machine learning-based methods. In threshold-based methods, fall events can be detected if the measured features values exceed pre-determined threshold values. Machine learning-based methods use labelled data to train a classier using supervised machine learning algorithms such as Support Vector Machines (SVM), Decision Tree, and Artificial Neural Networks (ANN) to recognise the characteristic features of falls. Figure 2.1 shows a diagram representing the phases of this research including phase 1: Data Collection, phase 2: Foreground Segmentation, phase 3: Features Extraction and phase 4: Fall Detection.

2.6 Discussion

In this chapter, a review of technologies and solutions for monitoring daily activities and automatically detecting falls was presented. These technologies are wearable sensor-based methods, ambient sensor-based methods and computer vision-based methods. Although the use of wearable sensors is popular in the human fall detection domain, there are some issues associated with their utilisation. For instance, wearable sensors are invasive as they require wearing and carrying various uncomfortable devices. In addition, fall detectors using acceleration data might generate a high number of false alarms. The knowledge gathered through this literature reveals that ambient sensor-based methods produce a high number of false alarms. Moreover, some of the ambient sensors require to be installed in the floor to cover the whole area of the monitored person.

For computer vision-based methods, there is no need for the older adults to wear an accelerometer enabled device and its use is not affected by ambient noise.

The review presented here provides an overview of the most cited approaches for fall detection. The fall detection methods can be classified into threshold-based methods and machine learning-based methods. The manually defined threshold values could generate false alarms since fall and non-fall activities could have similar acceleration. Machine learning-based methods are used as an alternative to reduce the number of false alarms.

In the next chapter, a description of data collection will be introduced. Video data recording is employed in this research in order to monitor ADLs of an elderly person and detect falls.

Chapter 3

Environment and Data Collection

3.1 Introduction

Human activities recognition and especially automatic detection of fall using computer-vision techniques can be useful for helping older adults living alone in a smart environment. Video-based fall detection techniques employ camera sensors to acquire video data. Then, various algorithms are used for detecting and tracking a person in the scene. In this context, the visual information extracted from consecutive frames of video data represents crucial information which helps to detect falls.

In order to evaluate the proposed automatic human fall detection approach, video data was recorded in a realistic home environment setting using a single camera. Additionally, the experiments were conducted using publicly available datasets; UR Fall Detection dataset and Le2i fall detection dataset [25]. In this chapter, the environment and the data collection system employed for this research are discussed. Various scenarios of the collected video data from a camera sensor to monitor the daily activities of an elderly person and to detect falls are also presented.

This chapter provides a detailed description of the acquired video data gathered for training and evaluation of various machine learning based algorithms for detecting falls. This chapter is organised as follows: in Section 3.2 details of collected video data including; experimental setup, the position of the camera and scenarios are presented. Section 3.3 presents details about the UR Fall Detection dataset. Section 3.4 gives information about Le2i fall detection dataset. Some conclusions are drawn in Section 3.5.

3.2 Data Collection and Video Recording

This section provides an overview of the collected video data acquired from a single camera system. The video dataset contains falls and normal daily activities acquired in realistic scenarios. The reason for collecting video data is to provide enough experimental data for this research. In addition, most available fall detection datasets used the same location for recording video data. That is, the same location is used for testing and training. To evaluate the robustness of the proposed method to changes in location, video data need to be collected in a realistic home environment and in several environments.

Data is obtained from visual recording of individual mimicking normal and unusual behaviour in an indoor environment. Data was acquired using a fixed/static camera. The information enclosed in the visual data is needed to enable the development of methods to automatically identify and distinguish between falls and daily living of occupants.

The video recording was conducted in a controlled environment where the participants performed normal day to day activities like standing, lying, walking and sitting. On occasions where the participant had to simulate falling, a well cushioned surface or sofa was used and the location was kept away from hard furniture. Before a participant was asked to simulate a fall, a first aider was present and for all other activities, a first aider was present within 5 minutes.

One of the major challenges for most fall detector approaches is the high rate of false alarms. Thus, the recorded data involved the simulation of falls and fall-like activities. Fall-like activities have similar characteristics to falls in terms of high motion, and a large vertical velocity which tend to trigger false alarms. Evaluating with video data including activities which generate high motion similar to falls can help to improve the performance of the proposal fall detection approach. Figure 3.1 represents the main steps for the collection of the video data. The collection of video data sequences include the following steps:

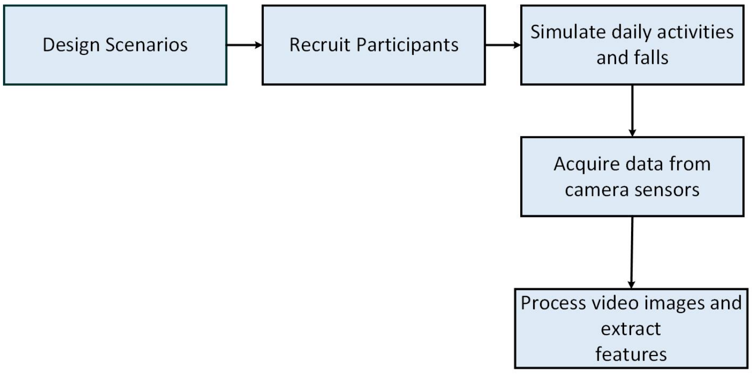

Figure 3.1: The data collection process.

- design scenarios for collecting video data sequences,
- identify appropriate hardware and camera sensors necessary for collecting data,
- recruit participants,
- collecting the data,
- process the collected data; detecting moving object and extract useful features. In this research study, various features are extracted from human silhouettes to evaluate the proposed fall detection algorithm.

After using a camera to record video sequences, the recorded data is used to extract features to determine physical shape, position and activity. This include three steps: the segmentation of foreground objects from the still background, object detection and feature extraction. Object detection methods are designed based on temporal information such as frame differencing and background subtraction. Object tracking is the process to track the person by locating their position in every frame of the video. After object detection and tracking, the next stage is to infer the behaviour of the occupant.

3.2.1 Experimental Setup

The experiments were conducted in a home environment, seven participants were invited to simulate fall and non-fall activities, which were used to test the pro-

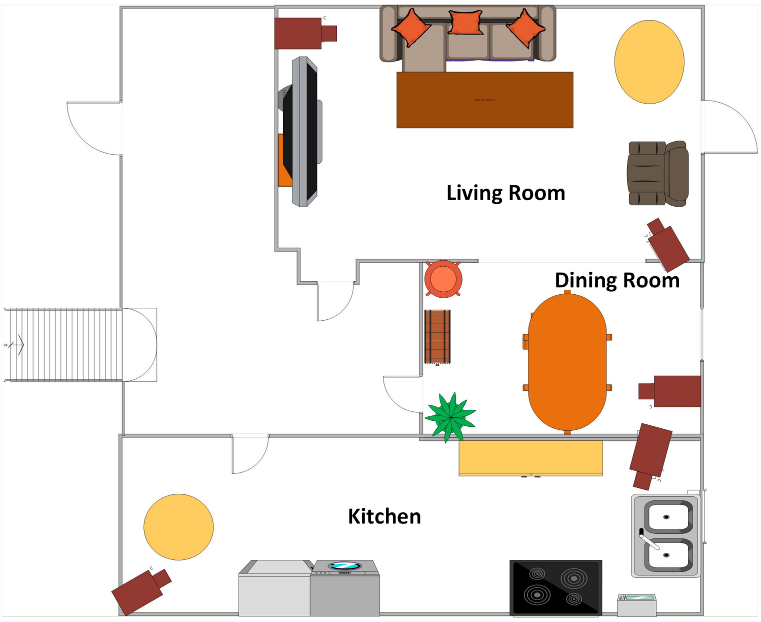

Figure 3.2: Layout of first floor of recording video data.

posed fall detection algorithm. The experiments were carried out in different rooms including living room, kitchen, dining room, office and bedroom with different light conditions. Participants were selected with varied age, gender and height. The data recording was conducted in a controlled environment and the participants performed normal day to day activities such as: walking, sitting, bending, lying on the sofa and also simulated falls. The video recording took place at the Crime Scene Training Facility at Nottingham Trent University, Clifton Campus, which is representative of the target environment and is comfortable for participants and those conducting the experiment. The ground floor has a kitchen, dining room and living room. On the first floor there are two bedrooms, office and bathroom. The layout of the house and the location of the camera sensors are shown in Figure 3.2 and Figure 3.3.

The data is collected using installed CCTV cameras with 640 x 480 pixels and the lens of the camera is 360 degrees to cover a good proportion of the room running at 30 fps. The distance of the person to the camera is approximately 3 meters. In addition, low cost RGB camera (Panasonic camera; SDR-S7 CD/ SDHC card with the image size of 768 x 576 as shown in Figure 3.4) mounted at 45 degrees is also used. The RGB camera was placed in the corner of each room

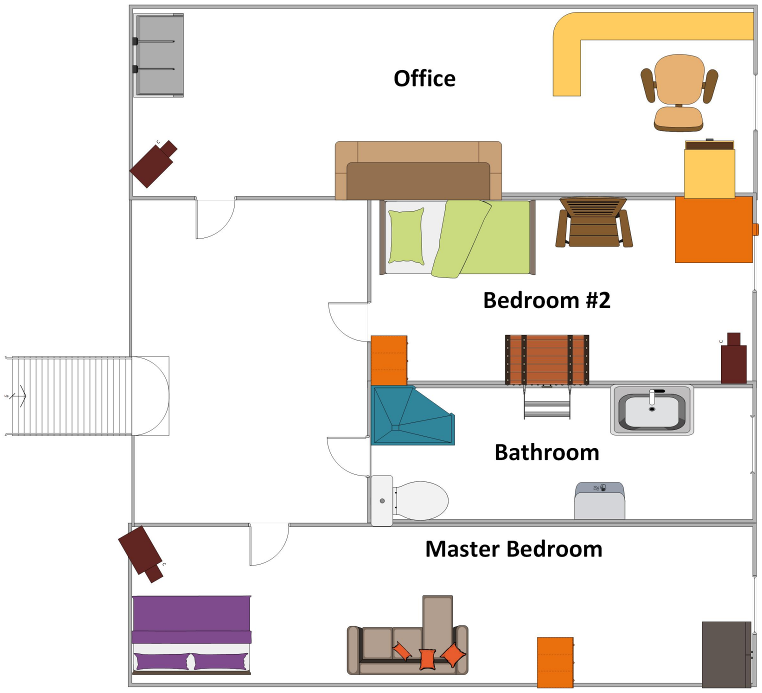

Figure 3.3: Layout of second floor of recording video data.

and attached to a tripod approximately 1.50 m above the ground. The videos are between 3 and 4 minutes long.

The recorded video sequence is processed by using MATLAB R2017b 64 bit on Windows 10. The Computer Vision System Toolbox and the Machine Learning Toolbox were used to implement the proposed fall detection system. The experiments were run on an PC laptop Intel (R) Core (TM) i5-4210U CPU @ 2.40 GHZ with 6 GB RAM.

Three different cameras were used to record the same scene at the same time, a ceiling mounted camera for side view (Camera #1), a ceiling mounted camera for front view (Camera #2) and the Tripod-RGB camera (Camera #3).

The recorded video sequences contained typical difficulties which can lead to segmentation errors mainly by the following reasons:

- shadows which can be detected as moving objects during the segmentation process,

- variable illumination,

Figure 3.4: RGB camera used for recording video data.

- occlusions caused by furniture like; chairs and table,
- entering and leaving the view of a camera,
- different clothes with different colours and textures.

A sample of images gathered from our experiments are shown in Figure 3.5.

3.2.2 Scenarios

The data recorded includes five different scenarios representing daily activities that may occur in an older adult's life. All these scenarios have only one person in the scene. Falls were simulated in different directions with respect to the camera view. Different types of fall incidents were recorded to include forward, backward and sideways falls. A fall event takes about 3 seconds and each participant performed three falls. In addition, there are 10 fall videos depict fall in diverse directions. The five scenarios were developed to collect data suitable for evaluation of fall detection algorithms. Each scenario involves a set of activities and events performed by participants during data acquisition. Each participant performed five scenarios, for a total time of 15 minutes per participant.

It was assumed that in real-life, people engage to some activities such as reading books, or watching TV while maintaining various postures. Therefore,

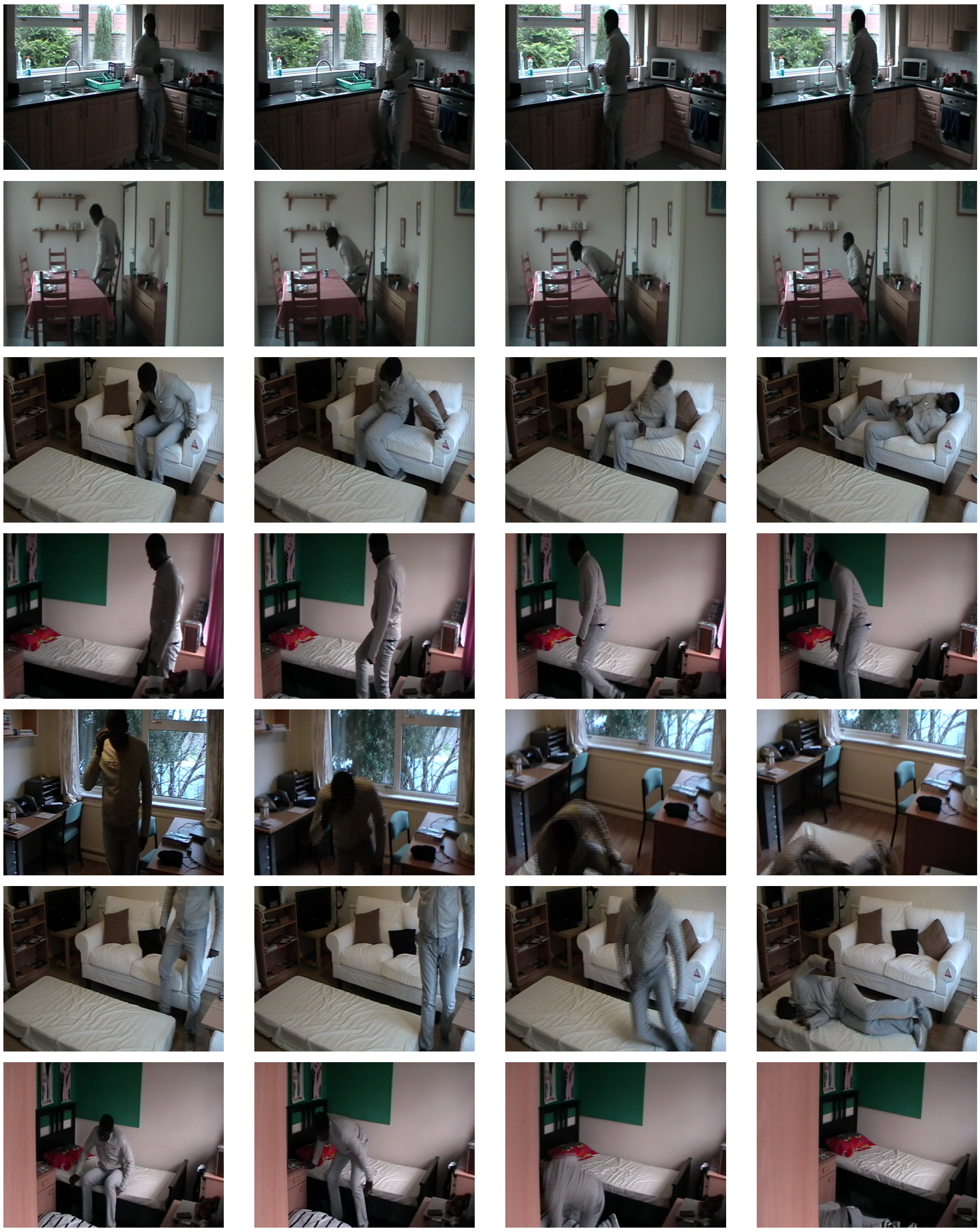

Figure 3.5: Sample frames for transition of normal daily activities including walking, sitting, lying down and bending are presented in the first four rows. Human falls in various ways side way fall, backward fall and occluded fall are presented in the last three rows.

in order to gather realistic ADL data, these activities were considered in our scenarios. For example, the ADLs of scenario 1 involve using a TV remote control and watching TV while the person is sitting on a sofa.

Falls are abnormal events that occur during normal daily activities. Therefore, the collected video sequences were developed to simulate falls during normal daily activities. Various scenarios define a set of normal daily activities during which falls can occur. For instance, in scenario 4, a person can be walking, then sitting down for a moment, then stand up and fall on the floor. Therefore, the collected data consist of a combination of static postures, transitions between different postures, and fall events. A survey done by [104] demonstrates that older adults tend to fall more when walking and when trying to get up from a chair or a bed. Thus, the recording video sequences include these types of fall incidents. Table 3.1 shows types of ADL and falls in the recorded video data.

According to Noury et al.[19] a fall event involves four phases:

- The pre-fall phase, corresponding to normal daily activities.
- The critical phase, corresponding to the fall event. This phase is extremely short, and can be detected by the high motion produced by a movement of the human body towards the ground.
- The post fall phase, where the person is in a lying position on the ground.

Table 3.1: Details of various activities in our recorded datasets.

Activities	Duration	Number of sequences
Sitting on a chair while reading a book	4 min	7
Sitting on a chair while drinking a cup of tea	3 min	7
Sitting on a sofa while watching TV	2 min	7
Laying on a sofa	3 min	7
Laying on a bed	4 min	7
Walking	1 min	35
Bending	15 sec	7
Fall forward	3 sec	10
Fall backward	3 sec	3
Side Fall	3 sec	17

This phase can be detected by an absence of motion.

- The recovery phase, where the person is able to stand up or sit down after a fall.

In order to perform fall events, participants were asked to deliberately simulate falls onto a 25 cm mattress and then to remain lying on the mattress with no motion. This process was performed for fall forward, fall backward, and fall side. Participants fell onto a mattress as realistically as possible. The following list gives the details of these scenarios.

- Scenario 1 sitting room: The participant is asked to sit comfortably in the sofa and watch Television for approximately 5 minutes. The person is then asked to walk close to the TV and pick up a remote, on the way back to the sofa, he/she is asked to simulate a fall. The simulated fall in this setup involves a mattress laid on the floor in the sitting room in-between the TV set and the sofa. When the participant is close enough to the mattress he/she is asked to simulate a fall.

- Scenario 2 kitchen: The participant is asked to walk to the sink, fill the kettle up with water and prepare a cup of tea. He/she is asked to walk to the sitting area and sit for approximately 3 minutes.

- Scenario 3 dining area: The participant is asked to sit down at the dining area for approximately 4 minutes and pretend to be having a cup of tea.

- Scenario 4 working area: The participant is given a newspaper and a story book to read at the desk. He/she is asked to walk around the bedroom for approximately 1 minute and then return back to the seat. After that, the participant is asked to simulate a fall, so he/she stands up and falls down to the floor.

- Scenario 5 bedroom: The participant is asked to walk around the bedroom for a minute and then lie down on the bed. This is followed by an uncontrolled fall on the floor such that the participant gets up from the bed and falls on the floor.

The main goal of activity recognition approaches is to be able to perform the recognition in videos coming from continuous streams of data. Given that, a dataset that includes long videos in which continuous streams of activities occur is needed [28]. Thus, for the fall detection dataset, considering more data in the period before and after the fall could improve the system performance. Thus, a feature vector extracted from a long video sequence will contain information in the period of fall and also information in the period before and after the fall.

3.2.3 Participants

A challenge was encountered during the acquisition of data relating to falls by the older adult. It is unethical to use older adults for performing falls due to the potential risks to their health [82]. Moreover, only few falls occur each year per older person. Thus, ADL and fall events in the acquired video data were simulated by young healthy participants. All participants gave their informed consent for inclusion before they participated in the study. Participants were healthy and independent, and none of them presented gait problems. Table 3.2 presents information about the participants recruited for the experiment.

The collected dataset included a total of 7 participants. Two females and five males were recruited. The ages of the participants was between 24 and 45 years, heights were between 1.58 and 1.86 m, and weight were between 56 and 98.5 kg. Age, weight, and height of the participants are provided in Table 3.2.

3.3 UR Fall Detection Dataset

In order to evaluate the efficiency of the proposed fall detection approach, experiments have been conducted using publicly available datasets; the UR Fall Detection dataset [62]. The UR Fall Detection dataset is widely used to test the

Table 3.2: Information about participants

Sex	Age	Height(m)	Weight(kg)
Male	$24-45$	$1.67-1.86$	$69-97$
Female	$30-37$	$1.58-1.65$	$58-78$

performance of various fall detection approaches in the literature.

3.3.1 Experimental Setup

The UR Fall Detection dataset includes front and overhead video sequences acquired by two Kinect cameras. The first camera mounted at a height of 1 m from the floor and the second camera placed at the ceiling with a height of 3 m. There are 30 fall scenarios recorded by two Kinect sensors and an accelerometer. The daily activities include: walking, sitting, bending, picking-up an object from the floor and crouching down. The normal activities (30 scenarios) were collected using one kinect sensor parallel to the floor, and 10 sequences with fall-like activities such as quickly lying on the floor and lying on the bed. Five participants were asked to perform daily activities and simulate falls. The dataset was recorded at 30 frames per second. The number of images in the video sequences containing fall events is equal to 3000, whereas the number of images in video sequences with ADLs is equal to 10000.

Falls were simulated in different directions with respect to the camera view. Different types of fall incidents were recorded to include forward, backward and sideways falls. Participants performed falls from standing position and from sitting on the chair. Frame samples taken from the UR Fall Detection dataset are shown in Figure 3.6. For the proposed fall detection approach, only video data from RGB camera are used.

3.4 Le2i Fall Detection Dataset

The Le2i fall detection dataset [25] contains 249 videos, 192 videos represent falls and 57 videos represent daily activities. The activities of daily living include walking in different directions, sitting down, standing up and crouching down. The video data were recorded in four different locations; home, coffee room, lecture room and office. The data was acquired by a single RGB camera placed 2m high from the floor. Participants attempted to simulate daily activities and falls

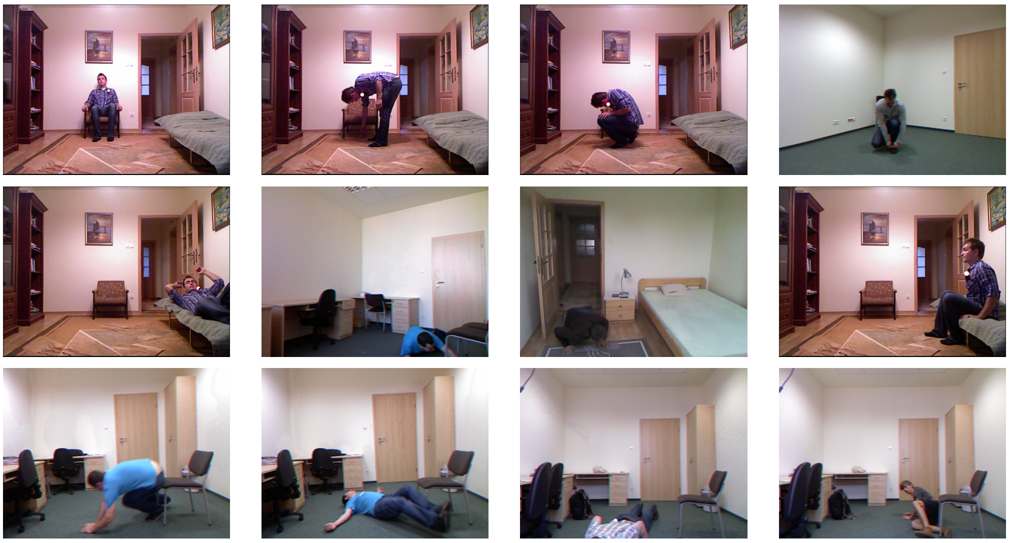

Figure 3.6: Sample frames from UR Fall Detection dataset. Upper row images are normal daily activities including sitting, lying down, crouching down and bending over. Second row images are fall-like activities including; laying on the floor and laying on the bed. Lower row images are human falls in various ways [62].

in different directions. A number of video sequences represent some difficulties such as; occlusions, cluttered, textured background, shadows and variability in illumination. The frame rate is 25 frame/s and the resolution is 320 x 240 pixels. Frame samples taken from the Le2i fall detection dataset are shown in Figure 3.7. Table 3.3 represents the characteristics of different fall detection datasets including; the UR Fall detection dataset, the Le2i fall detection dataset, and the collected video dataset.

After the data acquisition process, each video data is converted into a number of frames. Various computer vision techniques were applied to detect moving object and extract features from consequent frames. The acquired video data is used to extract the relevant features, providing the inputs for the fall detection methods.

3.5 Discussion

The collected video data set is intended for the evaluation of the proposed fall detection approaches by combining daily activities and falls. In the present study, to assess the detection performance of the proposed fall detection approach, experiments are conducted on the video data recording dataset, UR Fall Detection dataset and Le2i fall detection dataset. In this chapter, the experiments setup for data collection are discussed. Various scenarios for daily activities and falls are described in this chapter. In Chapter 5, 6 and 7, the collected data is used

Table 3.3: Characteristics of the different fall detection datasets.

Characteristics	Le2i fall detection dataset	UR Fall detection dataset	Recording video data
Normal activities	Normal activities like: walking and sitting but not lying down.	Some normal activities include: walking, sitting, lying, bending and crouching down.	Normal activities such as: walking, sitting, bending and lying.
Falls	Different kind of falls.	Several different types of falls.	Several different types of falls include person partly occluded.
Participants	Several participants wearing different cloths.	Several participants with different clothing.	Several participants with different types of cloths.
Lighting	Different light source, including sun light through window.	Different light source represents variable illumination.	Several light sources; changing conditions during day.
Setting	Several different environments include: living room, office, coffee room and lecture room, with few pieces of furniture.	Several different rooms include: living room and office, with only small amount of furniture.	Five different rooms involve: kitchen, dining room, living room, bedroom and office, with full of furniture.
Camera perspective	One perspective.	Two perspectives.	Fixed perspectives using two CCTV cameras placed upper corner of rooms. In addition, RGB camera fixed on tripod.
Occlusions	Person is partly occlusion.	Few video sequences represent person is partially in view of camera.	Occlusion caused by furniture and person exiting view of camera.
Video length	Short video sequences (10 – 45 sec).	Short video sequences (2 – 13 sec).	Video length between (3 – 6 min).

Figure 3.7: Sample frames from Le2i fall detection dataset. First and second row images are normal daily activities including standing, bending, and sitting down. Lower row images are human falls in various ways [25].

to train and test the proposed approaches for fall detection.

Chapter 4

Selected Features and Machine Learning Techniques for Fall Detection

4.1 Introduction

The proposed fall detection system includes four steps; data collection, foreground segmentation, feature extraction and fall detection. Background subtraction is implemented to segment out moving objects [94]. Afterwards, useful features such as motion information, shape orientation, temporal change of the head and projection histograms for detecting fall from different daily activities are extracted. The proposed system exploits motion, projection histograms and shape features based on the observation that human falls often involve drastic shape changes and abrupt motions as compared to other normal daily activities.

This chapter is organised as follows; Section 4.2 reviews various methods for detecting and tracking moving object. Section 4.3 presents enhanced features for the proposed fall detection approach. Machine Learning techniques for classifying fall and non-fall activities are investigated in Section 4.4. A brief discussion is presented in Section 4.5.

4.2 Detection and Tracking of Moving Object

Moving objects detection in video streams is one of the important research problems. Various techniques have been developed to detect and extract foreground object from a video sequence. Foreground object contains the interesting information to do further processing for several types of applications [119]. This section reviews the state of the art related to various techniques for detecting and tracking moving object in a video sequence.

4.2.1 Moving Object Detection

Identifying moving objects from a video sequence is a fundamental task in many computer vision applications [94]. The objective of moving object detection methods is to utilise a video sequence acquired from camera and produce a binary image representing moving objects for each frame of the sequence [119]. Jadhav and Jyoti [55] mentioned that most common methods for moving object detection are mainly the frame subtraction method, the background subtraction method and the optical flow method. This section briefly classifies various methods available for moving object detection from video.

4.2.1.1 Frame Subtraction Method

The work presented in [101] uses frame subtraction method to segment out the moving object from background image. Firstly, the first frame is captured through the camera. Secondly, the absolute difference is calculated between the consecutive frames and the difference image is stored in the system. Thirdly, the difference image is converted into binary image. Finally, morphological filtering process is used to remove noise [101]. According to [31] the frame subtraction method is simple in implementation and has low computational requirements. However, it is not suitable for detecting darker objects over/on a light background. Additionally, the main challenge of this method is the determination of an appropriate threshold, since the result depends on the threshold used and lighting conditions [31]. The main limitation of the frame difference method is that if the object moves very slowly it might not be detected.

4.2.1.2 Background Subtraction Method

The background subtraction method uses the difference between the current image and the background image to detect moving objects. Various stages of the background subtraction method are shown in Figure 4.1. The background image is subtracted from the current frame. Any pixels with difference greater than the set threshold can be classified as foreground pixels. The value of threshold can change dynamically using the dynamic threshold method. This method can effectively support the impact of light changes [55].

The basic idea of the background subtraction method is to initialise the background with the first input image. The subsequent images are then converted to gray-scale and then subtracted from the background image at the pixel level to produce a binary image as in (4.1) [119].

$$R_k(X,Y) = F_k(X,Y) - B(X,Y). \tag{4.1}$$

R_K is the absolute difference between the current and background images; F_K is the current image and B is the background image. If the value of pixel level difference is lower than the threshold value, the object is considered to be background pixel and assigned 0 in the binary image. Otherwise the pixel is considered as foreground and assigned 1 as in (4.2) [55].

$$D_K(X,Y) = \begin{cases} 0 & background \quad R_k(X,Y) \geq T, \\ 1 & target \quad R_k(X,Y) < T. \end{cases} \tag{4.2}$$

There are many background subtraction methods used for segmentation for instance: mean filter, W4 and inter-frame difference [96]. The mean filtering method is calculated using the mean of the last n frames. This is a fast, and easy to implement algorithm. Mean filtering also uses an adaptive background calculation. The drawbacks are that the accuracy of this method depends on object speed and the memory requirement is very high [96]. Inter-frame difference method subtracts the corresponding pixel value of the front from that of the back. If the result is less than a certain threshold, there is no moving object target otherwise; there is a target [67].

The work done by [114] proposes a new inter-frame difference algorithm for

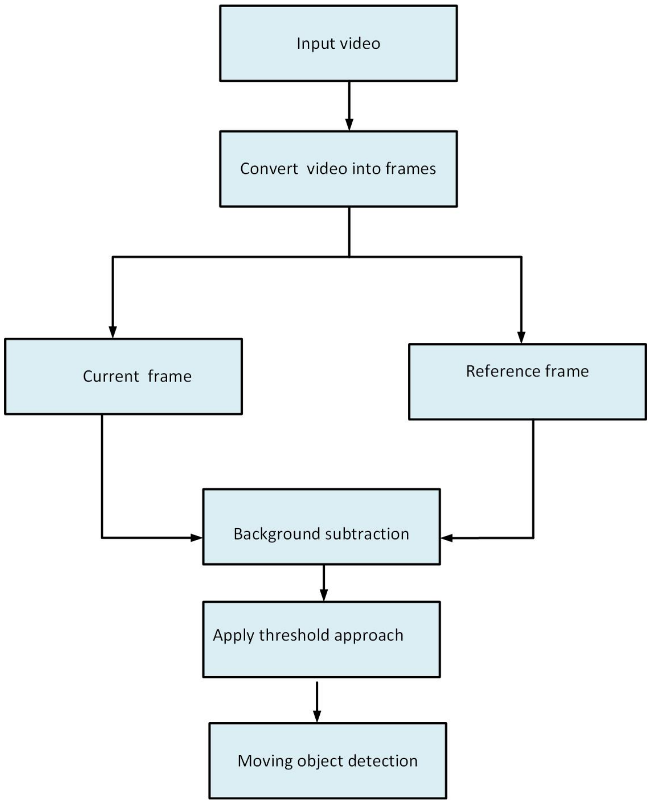

Figure 4.1: Background subtraction method.

moving target detection. This algorithm is based on three- frame- difference methods in combination with the background subtraction method. The optical flow method detects the optical flow change of each pixel in the image to distinguish the moving target from the background. This method can detect the moving object from the background without the effect of motion background [35].

4.2.1.3 Statistical Methods

Statistical methods have been utilised to overcome the shortcomings of background subtraction techniques. Statistical methods are mainly based on dynamically updating statistics of the pixels belonging to the background image [119]. Some studies based on using statistical methods such as Gaussian Mixture Model (GMM) and Hidden Markov Model (HMM) for detecting moving objects in video sequences are presented below.

Gaussian Mixture Model (GMM) [84] uses the moving object distribution in the first frame of the video sequences to localise the object in the next frames by tracking its distribution. The GMM has been used widely for modelling dynamic background as it can represent a complex distribution of each pixel. Moreover, GMM is computationally intensive and very sensitive to sudden changes in global illumination which can turn the entire frame into foreground [94]. The GMM is a complex technique because it uses a different threshold for each pixel and has many parameters to update, which reduce its speed. These limitations can be overcome using the Approximated Median Filter (AMF) that achieves good performance with simple implementation. However, it is slow to adapt to larger change in background. The AMF needs many frames to learn the new dark background [96]. Median filtering is a nonlinear single processing method. It is widely used to remove noise from images while preserving edges. The median filter works by replacing each pixel value with the median value of neighbouring pixels. The pattern of neighbours is called the window. The median is computed by sorting all the pixel values from the window into numerical order, and then replacing the pixel being considered with the middle pixel value [129].

Hidden Markov Model (HMM) is widely used for object background subtraction. It represents the intensity variations of a pixel in an image sequence as discrete states. The HMM can be used in the context of detecting light on/off events in a room [35].

4.2.1.4 Optical Flow Methods

Optical flow methods uses the flow vectors of moving objects over time to detect moving regions in an image. In this approach, the apparent velocity and direction of every pixel in each frame are computed. This method is effective to detect motion in video sequences even from a moving camera and moving background. However, most of the optical flow methods are computationally complex and time consuming [119]. Because the sensors that will be used here will mainly be stationary there will be no added advantage to consider this computationally intensive approach.

4.2.1.5 Morphological Operations

Moving object detection is not an easy task due to many challenges involved in detecting moving objects in video sequences captured by cameras. Some of the challenging issues includes: sudden changes in the speed and direction of the objects motion, occlusion, shadows and reflections [119]. Post processing techniques are widely applied to remove these false errors in the output images. Morphological operations and connected component labelling were used as post processing techniques to remove false errors and refine the edge of resulting moving object on the binary image. The fundamental morphological operations are listed below.

- Dilation operation is basically used for filling the holes in a continuous object. The dilation operation gradually enlarges the boundaries of foreground regions. Therefore, areas of foreground objects become larger while holes within those areas become smaller.

- Erosion operation is the complement of the dilation operation. That is erosion operation results in loss of boundaries of foreground regions. Thus, areas of foreground objects shrink in size, and holes within those regions become larger.

- Opening operation is defined as an erosion followed by a dilation. Opening operation generally smoothes the outline of foreground objects by removing some of the foreground pixels from the edges of foreground regions.

- Closing operation is defined as a dilation followed by an erosion. Closing operation smoothes sections of contours, blends narrow breaks and fill gaps in the boundaries of foreground regions [88].

4.2.2 Moving Object Tracking

A video tracking system usually has three stages: object extraction, object recognition and tracking, and high-level decisions about the object [85]. The aim of an object tracker is to recognise the motion trajectory of an object as video frames progresses by identifying the objects position in every frame.

Tracking moving objects depends on many features which describe the appearance of the objects. These features may include edges, colour, gradient, texture, optical flow and Spatio-temporal features. Colour feature descriptors are used to increase the discriminative power of intensity based descriptors. Gradient features are important in human detection. For example, the shape or contour of the human body is used in gradient based methods to represent the human body. In [117] a contour based object tracking algorithm was implemented to track object contours in video sequences.

Firstly, the active contour is segmented using the graph cut image segmentation method. The resulting contour of the previous frame is taken as initialisation in each frame. The new object contour is detected using intensity information of the current frame and the difference between the current frame and the previous frame. According to [26] contour based methods can achieve a high tracking precision; however, the computing cost of these methods is usually high, especially for large and moving objects. There are various contour tracking approaches such as snakes active contour models, which are effective in extracting object contours and tracking them in the video [117]. Using region-based object tracking models, objects are tracked by considering the colour distribution of the tracked object. However, these methods are not suitable when multiple objects move together in the image sequences. The basic idea of region based method is to track objects with the similarity measure of object region. The new similarity measure in the spatial feature space can effectively cope with the translation and scaling of the object. However, it does not consider the rotation invariance [26].

Kalman filter is a set of mathematical equations that provide an efficient computational means to estimate the state of a process [26]. The Kalman filter can be defined as an optimal recursive data processing algorithm composed of two stages: prediction and correction. The first stage includes the prediction of the next state variable using the current set of observations. The second step gradually updates the predicted values and gives a much better approximation of the next state [94]. The Kalman filter stages are shown in Figure 4.2. The Kalman filtering approach can be used to track points in noisy images [16].

After detection of moving object by background subtraction algorithms, the next step is to track the extracted the human silhouette from frame to frame to

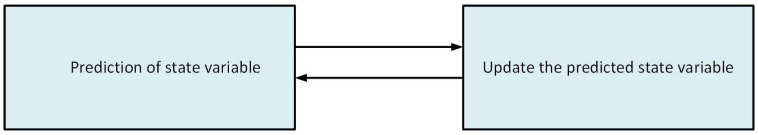

Figure 4.2: The basic stages of Kalman filter.

analyse behaviour. Tracking human silhouette is done by extracting combination of some features from the shape of the human silhouette. The ability to distinguish a fall action depends mainly on the quality of the classifier input, therefore, the features of the extracted human silhouette play a key role in the effectiveness and robustness of detecting human falls [110].

The next section will introduce the implementation of methods to extract these features covered by this research.

4.3 Enhanced Features for Fall Detection

The feature extraction process includes applying background subtraction algorithms to extract the human silhouette, and combination of some features from the shape of the human silhouette. These features retain the motion information of actions and includes the motion, changes in orientation, changes in head position and histogram features are extracted from the human silhouette. The extracted features are used to decide whether there is a fall or not. This section presents details on how the features are extracted.

The first step of the system is to analyse the motion occurring in a given time window, using the tMHI [66]. The proposed method is based on the assumption that the motion is large when a fall occurs. Based on this assumption, the proposed system aims to detect a large motion of the person in the video sequence using tMHI. The motion is quantified by calculating the pixel value of motion history image blob in the current frame, which is then divided by the number of pixels in the human blob. A large motion does not necessarily signify a fall, as activities like fast walking or running may also exhibit such characteristics [95].

Considering that the first step is not sufficient, a second step is also considered to analyse the change of the human shape to identify a fall among other activities. Analysis of the moving object is performed by fitting an approximate ellipse around the human body. The orientation of fitted ellipse provides information about the body posture [41]. After ellipse fitting, the orientation of the ellipse and the ratio between the major semi-axis a and the minor semi-axis b are taken as features to describe a human body posture in a general way. However, these features alone are not sufficient to describe postures in detail for distinguishing different postures [121], therefore more features are needed.

It is assumed that human fall have greater motion than other daily activates like walking or sitting. However, focusing only on a fast motion can result in many false alarms during fall-like activities like sitting down quickly [61]. Therefore, combining motion with other features extracted from the fitted ellipse around the human body helps to discriminate actual fall from other activities. After ellipse fitting, the orientation, the ratio, the major semi-axis and the minor semi-axis are taken as a feature to describe a human body posture. The motion feature C_{motion} indicates the changing rate of human motion and orientation feature θ indicates the changing rate of human shape [9].

Common fall detection algorithms are based on the fact that the fall activity have high acceleration than other normal activities. This study highlights the situation when human fall can occur with low motion rate. A typical example is when a person loses balance and holds onto a furniture to prevent a fall and yet falls on the ground. Therefore, the third feature which is the projection histogram feature is applied to confirm a fall event. Furthermore, the proposed fall detection approach applies another feature; tracking of the human head in subsequent frames to deal with occlusion problem. Occlusion can occur when a relevant area of the bottom of a person is covered or when a person moves behind an object and consequently part of his/her body disappears [32]. Tracking the head position of the person can provide useful information in such instances as they tend to be visible most of the time. Also in a fall situation there is more movement toward the head than the lower part of the body. The bounding box was used to surround the foreground object, then the y-coordinate of the top left point of the bounding box is computed and the absolute difference of y-coordinates in successive frame are used as features to fed into various machine learning algorithms for fall and non fall events classification.

4.3.1 Motion History Images

Motion History Image (MHI) has been used widely to extract the motion from video sequences. The MHI indicates the speed of the movement; therefore, if a person conducts unusual activities such as fast walk or run, this method will return a high value as the result. In [116], MHI was used to extract motion from the video sequences. After that, the standard deviation of (a) the motion quantification and (b) orientation of the ellipse were employed to distinguish a fall from other activities; for instance, walk, sit and lie.

The MHI method is an approach based on template matching. The MHI method can provide useful motion information of a moving object and it is insensitive to illumination change and occlusion. These advantages make the MHI method suitable for motion analysis in challenging scenarios [66].

Suriani and Hussain [105] employed two motion features namely, Motion History Histogram (MHH) and Motion Geometric Distribution (MGD) to distinguish the transition state between walking and falling. The geometric distribution of motion can be used to identify spatial abnormalities in an event such as a sudden fall. In addition, MHH is used to differentiate between other multiple actions like walk, run, sit and fall.

The work of [116, 105, 8] highlighted that the MHI method is simple to calculate and has been employed widely for various action-recognition tasks. Although, MHI is easy to implement, it has the limitation of viewpoint dependence and loss of information in the projection from 3D to 2D [8]. Indeed, [7] points out that the main restrictions of MHI are the motion self-occlusion and motion overwriting which lead to loss of important information. Additionally, MHI is dependent on the temporal duration value and it cannot perform well with variable length action sequences.

To overcome some of the constrains of the MHI method, a variant of the MHI, like the timed Motion History Image (tMHI) method can be used. In [20], the tMHI method is employed with silhouette pose recognition which provides a potentially useful tool for gesture and motion recognition. In addition, the tMHI is used by [95] for motion segmentation to track objects in real time. In this study, the tMHI method will be used for motion segmentation. This method makes the representation independent of the system speed or frame rate, so that tMHI can cover the same MHI area at different capture rates. Detailed discussion as to why this method was chosen will be addressed in the Section 4.3.1.1.

4.3.1.1 Timed Motion History Image

The MHI is generalised by directly encoding the actual time in a floating point format, which is called timed Motion History Image (tMHI) [95]. With tMHI, representation can be used to determine the current pose and also measure the motion of an object. In general, the tMHI is updated by time stamps of the video sequence, rather than the frame numbers [8]. This method makes the representation independent of the system speed or the frame rate. One can conclude that tMHI provides coherent motion information to represent the motion trail of a moving object over time [66]. The motion history image contains the trajectory information of the action being performed and recent motion is emphasised more than past motion [48].

When the human body region is extracted, the motion activity of the segmented foreground object is measured by generating a timed motion history image (tMHI). The tMHI image is computed as:

$$tMHI_\delta(x,y) = \begin{cases} \tau & if \quad current \quad silhouette \quad at(x,y). \\ 0 & if \quad tMHI_\delta(x,y) < (\tau - \delta). \end{cases} \tag{4.3}$$

where τ is the current time-stamp, and δ is the maximum time duration constant (typically a few seconds) associated with the template [8, 95].

Compared to tMHI, MHI refers to an image where the pixel intensity represents the recency of the motion in the image sequence, and therefore gives the most recent movement of the person during an action [8]. A MHI image has the same size as the input image and contains motion information associated with any action in the frame. A Motion History Image ($H_{\tau(x,y,t)}$) at time t and location (x,y) is defined by the following equation:

$$H_\tau(x,y,t) = \begin{cases} \tau \quad if \quad D(x,y,t) = 1, \\ H_\tau(x,y,t-1) \quad , otherwise, \end{cases} \tag{4.4}$$

where $D(x, y, t)$ is a binary sequence of motion regions which is obtained from the original image sequence using background subtraction method. Each pixel of Motion History Image H_τ is a function of the temporal history of the motion at that point occurring at fixed duration τ (with $1 \leq \tau \leq N$ for a sequence of length N frames) [8].

4.3.1.2 Quantify the Motion

To quantify the human motion, it can be calculated using pixel values of motion history image, divided by the number of pixels in human blob. The coefficient of motion C_{motion} based on the motion history image can be computed as shown below:

$$C_{motion} = \frac{\Sigma pixel(x, y) \in blob \ H_\tau(x, y, t)}{\#pixels \in blob}. \tag{4.5}$$

A coefficient C_{motion} is computed based on timed motion history image using:

$$C_{motion} = \frac{\Sigma pixel(x, y) \in blob \ tMHI_\delta(x, y)}{\#pixels \in blob}. \tag{4.6}$$

The term *blob*, refers to the silhouette of a person extracted using the background subtraction method, and $tMHI_\delta$ means the timed Motion History Image. The value of C_{motion} is a percentage ranging from 0% (no movement) to 100% (maximum movement) [92].

4.3.2 Approximated Ellipse

One of the conventional methods for detecting a fall from video sequences is to represent the human shape using the bounding box method. This method is simple and easy to implement [87]. The bounding box attributes of height and width are used to represent the human physical shape. The approach proposed in [33] extracts four features from the bounding box around the human silhouette to describe a fall. The features are the aspect ratio, fall angle, centre speed and head speed. A Support Vector Machine classifier is then employed to detect the fall using these features. However, The major drawback is the inadequate description of human motion by simply using a bounding box [122].

The other effective model is to represent the human in the video using an ellipse. The study in [46] presents a novel method to detect falls which combines the orientation angle and the ratio of a fitted ellipse around the human body, motion coefficient and silhouette threshold features. The extracted features are then used as inputs to a KNN classifier to classify fall events. The accuracy of the system was 95%. The authors in [25] used fourteen features extracted from the bounding box including height, width, aspect ratio, centroid coordinates of the box and ellipse orientation for fall detection. Fourier and wavelet transformations were then applied to these features before fall detection by using either SVM or adaptive boosting algorithm. Using the Le2i fall detection dataset, they achieved specificity of 100%, an accuracy of 99.9% and recall of 98%. The work presented in [69] proposes a fall detection approach using a Gaussian mixture background model to build the background. MHI is applied to analyse the fall behaviour and the orientation and the ratio of the ellipse are computed to represent the variation in the shape of the human object. In addition, two extra features, acceleration and angular acceleration, are computed to improve fall detection accuracy. However detailed performance data was not provided.

The ellipse model is a simple model describing the motion or the shape of the human body. In this model, a single object is surrounded by an ellipse. The approximated ellipse offers information in relation to the shape and orientation of the person in the image [41]. There are three important parameters of the ellipse: a) the vertical angle of the object (current angle), b) the major axis of the object and c) the minor axis of the object. In [120] , analysis of the moving object is performed to detect a change in human shape; particularly in orientation and proportion.

The moving object is approximated by an ellipse using a moment-based method [44]. The general form for calculating the moment of any two variable functions is given as:

$$m_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p \ y^q \ f(x,y) dxdy, \tag{4.7}$$

for $p, q = 0, 1, 2...$ and binary image $f(x, y)$, the moments of order $(p+q)$ is given by:

$$m_{pq} = \sum_{x,y} x^p \ y^q f(x,y) \ with \ p,q = 0,1,2,3\ldots \tag{4.8}$$

To calculate the region of interest in a binary image $f(x,y)$, one will have to calculate its zeroth moment as:

$$m_{00} = \sum_{x,y} x^0 \ y^0 \ f(x,y). \tag{4.9}$$

Therefore, the x^0 ,y^0 do not have any effect and can be removed. $x^0 = 1$ and $y^0 = 1$. Thus, the zeroth order moment, m_{00} of the image $f(x,y)$ can be defined as:

$$m_{00} = \sum_{x,y} f(x,y). \tag{4.10}$$

The above equation counts all the white pixels in an image. Therefore, the zeroth moment represents the total area of the size of the image. To compute the centre of the ellipse (x,y), the first and zero moments, m_{10}, m_{01}, m_{00} are calculated as:

$$m_{10} = \sum_{x,y} x^1 \ y^0 \ f(x,y), \tag{4.11}$$

$$m_{10} = \sum(x). \tag{4.12}$$

The x is the coordinate of all white pixels (where $f(x,y) = 1$) is summed.

$$m_{01} = \sum_{x,y} x^0 \ y^1 \ f(x,y), \tag{4.13}$$

$$m_{01} = \sum(y). \tag{4.14}$$

The y is the coordinate of all white pixels (where $f(x,y) = 1$) is summed. After calculating the sum of the x and y coordinates of several pixels, the average is computed by dividing each moment by the number of pixels in the image.

$$\bar{x} = m_{1,0}/m_{0,0} \ , \ \bar{y} = m_{0,1}/m_{0,0}. \tag{4.15}$$

The coordinates x and y of the centre of the image are described by the spatial moments of first order m_{10} and m_{01} divided by the zero order moment m_{00}. After that, basic moments of order 0 to 2; m_{01}, m_{10}, m_{20}, m_{02}, and m_{11} are calculated as follows:

$$m_{20} = \sum_{x,y} x^2 \ y^0 \ f(x,y), \tag{4.16}$$

$$m_{02} = \sum_{x,y} x^0 \ y^2 \ f(x,y), \tag{4.17}$$

$$m_{11} = \sum_{x,y} x^1 \ y^1 \ f(x,y). \tag{4.18}$$

These moments are then used with the centroid $(\bar{x}, \bar{y})$ to compute **the central moments** μ_{11}, μ_{20} and μ_{20} as follows:

$$\mu_{pq} = \sum_{pq} \ (x-\bar{x})^p \ (y-\bar{y})^q f(x,y), \tag{4.19}$$

with $p, q = 0, 1, 2, 3...$, therefore:

$$\mu_{11} = \frac{m_{11}}{m_{00}} \ - \bar{x} * \bar{y}, \tag{4.20}$$

$$\mu_{20} = \frac{m_{20}}{m_{00}} \ - \bar{x}^2, \tag{4.21}$$

$$\mu_{02} = \frac{m_{02}}{m_{00}} \ - \bar{y}^2. \tag{4.22}$$

The central moments μ_{pq} of the image is required for computation of the orientation of the ellipse. The angle between the major axis of the person and the horizontal axis, gives the ellipse orientation and can be computed with the central moments of second order. The orientation describes the direction of the major axis and is within the range $-\pi/4 \leq \theta \leq \pi/4$ and it is defined as:

$$\theta = \frac{1}{2}\, arctan(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}). \tag{4.23}$$

The eigenvalues I_{min} and I_{max} are given by:

$$I_{min} = \frac{\mu_{20} + \mu_{02} - \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}}{2}, \tag{4.24}$$

$$I_{max} = \frac{\mu_{20} + \mu_{02} + \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}}{2}. \tag{4.25}$$

Then, the major semi-axis, a, and the minor semi-axis, b, of the best fitting ellipse are given by:

$$a = (4/\pi)^{1/4} \left[\frac{(I_{max})^3}{I_{min}}\right]^{1/8}, \tag{4.26}$$

$$b = (4/\pi)^{1/4} \left[\frac{(I_{min})^3}{I_{max}}\right]^{1/8}. \tag{4.27}$$

The ratio of the ellipse is computed by $\rho = a/b$ [41].

Fitting an ellipse is insufficient to describe the posture of the human body in detail and it might be hard to differentiate two postures by using only the global information [121]. Therefore more information from local features is required to describe different postures. A widely used feature to describe such detailed information is the projection histogram. The projection histogram features are computationally efficient to derive and produce a good performance for posture classification [120, 70]. The work in [120] presents fall detection using the information from ellipse fitting and a projection histogram along the axes of the ellipse to distinguish different postures of the person. The system achieves a high fall detection rate of 97.08% in a simulated home environment. Similarly, the approach presented in [123] is based on measuring a temporal variation of pose change and body motion to detect falls. Several measures such as centroid velocity, head-to-centroid distance, a histogram of oriented gradients and optical flow were computed. The system can correctly classify 90.6% of falls. A comparison amongst several fall detection systems was performed in [22], showing sensitivities from 71% to 100%, specificity from 73% to 100% and accuracy of 84 to 94%.

4.3.3 Projection Histogram

The horizontal and vertical projection histogram of foreground object is obtained by calculating the number of foreground pixels row wise and column wise. The foreground F image is donated as cloud of $2D$ points with (x_p, y_p) as the pixel coordinates. The horizontal projection histogram $Hz_(y)$ of foreground F can be defined as cardinally of set of points as follows:

$$Hz_{(y)} = |(x_p, y_p) \in F, (y_p = y)| \,. \tag{4.28}$$

Similarly, vertical projection histogram $Vt_{(x)}$ can then be computed as follows:

$$Vt_{(x)} = |(x_p, y_p) \in F, (x_p = x)| \,. \tag{4.29}$$

For each activity, the horizontal and vertical projection histograms are computed for each frame. Then, the maximum values of foreground pixels in the horizontal and vertical histograms, as well as the difference between maximum values are calculated to effectively discriminate fall among other activities [43].

4.3.4 Temporal Changes of Head Position

The reason to track the head is mainly because it is visible in the scene and it has a large movement during the fall. When the fall occurs the head moves abruptly and its displacement would be large, thus, it is aimed to estimate a person's head position in every frame of a video sequence. In order to determine the head position, the top left detected point of silhouette is marked. Firstly, silhouette is enclosed by minimum bounding box and then the top left detected point of the bounding box is marked in each frame. In addition, the absolute difference values of top left point of the head over successive frames are obtained and forms feature vector, which represents the vertical displacement of the head point [43].

In the moment when a fall occurs, the y-coordinate of the person's head increases significantly, which leads to considerable variance in the vertical velocity. Consequently, the y-coordinate of the person head was selected as a feature for fall identification. In addition, the standard deviation of y-coordinate σ_y is calculated as:

$$\sigma_y = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left|y_i - \mu_y\right|^2}, \tag{4.30}$$

where y_i is the y coordinate which is calculated from the i^{th} frame; μ_y represents the average value of y within the specified number of frames; N is number of frames [116].

The absolute difference of consequences y-coordinates are also used as a feature for fall detection. The difference in y-coordinate is computed as:

$$d_y = y_i - y_{i-1}, \tag{4.31}$$

where y_i is the y coordinate of human head in the ${}^{i}th$ frame and y_{i-1} is the y coordinate of the human head in the $(i-1)^{th}$ frame. Moreover, the standard deviation of absolute difference of y-coordinate of the head is calculated using (4.30).

4.3.5 Selected Features

Based on the information provided in the preceding sections, for each video sequence, the moving object is detected and 10 unique features represented in the feature vector F are used to identify falls. The selected feature vector is given by:

$$F = [C_{motion}, \theta, \rho, a, b, Hz_{(y)} - Vt_{(x)}, y, \sigma_y, \left|y_i - y_{i-1}\right|, \sigma_{|y_i - y_{i-1}|}], \tag{4.32}$$

where the individual features are as follows:

- coefficient of motion C_{motion} - defined in (4.6),
- the orientation of the ellipse θ - defined in (4.23),
- the major semi-axis a,
- the minor semi-axis b of the ellipse and,

- the ratio of the ellipse ρ - i.e. the ratio of a/b where a and b are defined by (4.26) , (4.27),
- the difference between the horizontal and vertical projection histograms $Hz_{(y)} - Vt_{(x)}$ - which are defined by (4.28) and (4.29),
- the y-coordinate of the head point y,
- the standard deviation of y-coordinate σ_y - defined by (4.30),
- the absolute difference of y-coordinate d_y - defined by (4.31),
- the standard deviation of the absolute difference of y-coordinate $\sigma_{|y_i - y_{i-1}|}$ - defined by (4.30) as applied to d_y.

4.4 Machine learning techniques

In this section, some of the machine learning techniques used in fall detection are reviewed. These techniques are used later on in this thesis in Chapter 6 and 7 for fall classification. The utilised techniques are briefly described in this section.

4.4.1 Artificial Neural Network

Artificial Neural Network (ANN) has been applied widely on human pattern recognition due to its capability to learn from data and create a network model. The network model can be applied on new data which was not previously exposed to the network for classification. The ANN is a model consisting of a collection of inputs and processing units called nodes or neurons. The neurons are arranged into three layers: input layer, hidden layer and output layer. Each neuron performs the simple operation to process these inputs and produce the output and then the output is forwarded to the next node or neuron in the sequence[108]. The neurons, which consist of an activation function $\phi(W - T)$, where W is the weighted sum of the inputs and T is the bias value. The weights are initialised to small random values and updated during the training process. The weighted sum W is given from inputs 1,2,3,... to n and associated weights as shown in (4.33).

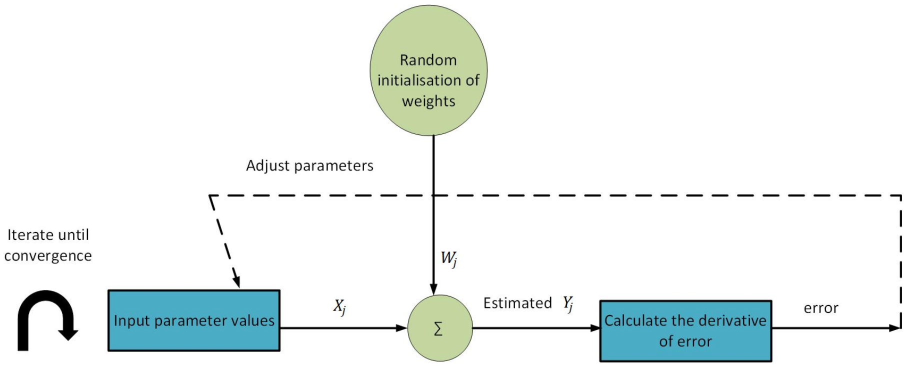

Figure 4.3: The learning process of a single neuron of the neural network.

$$W = \sum_{i=1}^{n} weight_i \times input_i. \tag{4.33}$$

A number of input vectors are provided to the algorithm in order to determine the corresponding desired output. There is an error created at the output layer when the input data is presented to the system. The error represents the difference in value between the real system output and the desired response value. The error is fed back into the ANN system to adjust its weights through the use of a learning rule [11].

During the learning phase, a set of training input vectors are presented at the input layers which are feature vectors and their corresponding desired output vectors. Initially random weights are assigned to the set of nodes. The ANN adjusts the weights attached to the connections according to the difference between the network's output and the desired output for that input vector. The more this difference is reduced the better for the classification outcome. The learning process of a single neuron is shown in Figure 4.3.

A single neuron in the network can be represented as follow:

$$Y_j = f_j \sum W_{ij} \times x_i, \tag{4.34}$$

where x_i is data input to the neural network, W_{ij} represents weights between i^{th} neuron of previous layer and j^{th} neuron of the current layer and f_j represents the activation function. There are various activation functions including: linear, sigmoid and hyperbolic tangent [124].

The ANN used in this study are based on the principles of the Multi-Layer Perceptron (MLP) network with Back Propagation (BP) learning algorithm because it is easy to train and for its accuracy. The process of the BP can be divided into two operations, feed forward and back propagation operations. In the feed forward operation, input features are fed to the input neurons. Then the back propagation adjust the output through adjusting the weights of the network [108].

A cross validation method is applied to estimate and evaluate the performance of the learning model. Section 4.4.5 presents various cross validation methods. Several training algorithms for ANN are based on the conjugate gradient algorithms. In this research, a variation of the conjugate gradient algorithm known as Scaled Conjugate Gradient (SCG) algorithm is used to train the feed forward neural network. The next section briefly introduces this algorithm.

4.4.1.1 Scaled Conjugate Gradient Algorithm

The learning process of the ANN often involves adjustments of weights. Optimisation methods such as conjugate gradients that are applicable to large scale problems are used as alternatives to the learning algorithms. The SCG is a supervised learning algorithm based on a class of optimisation techniques known as the Conjugate Gradient methods. The SCG is faster than the standard backpropagation algorithm. The SCG avoids time consuming line search per learning iteration by using the step size scaling mechanism, which makes the SCG algorithm considerable faster than the other second order. The minimisation strategy of the SCG algorithm is a local iterative process in which an approximation to the function in a neighbourhood of the current point in weight space is minimised.

The training algorithms based on the gradient descent methods usually have a poor convergence rate and depends on parameters specified by the user. The values of these parameters are often crucial for the success of the algorithm. The SCG algorithm combines Levenberg-Marquardt algorithm with a conjugate gradient approach.

In our experiments, the neural network is trained using the SCG Back-propagation algorithm. The SCG algorithm selected is fast to train and even with few training data, it is capable of generating acceptable results [79].

4.4.2 Support Vector Machine

The Support Vector Machine (SVM) algorithm is widely applied for classifying data [29]. The principle of the SVM algorithm is to map the input space into a higher dimensional feature space using a kernel function and then define the optimal separating hyperplane in the transformed space to distinguish between classes [23]. The optimal hyperplane for the SVM means, the one with the largest margin between the two classes, so that the distance to the nearest data point of both classes is maximised. Such a large margin means the maximal width of the tile parallel to the hyperplane that contains no interior data points and thus incorporating robustness into the decision making process [125].

The support vectors are the training data points that are closest to the separating hyperplane; These points represent the maximum-margin hyperplane for the training data. The SVM algorithm finds the optimal separating hyperplane to map each feature vector into its corresponding label space. Figure 4.4 shows the SVM maximum-margin hyperplane.

Different kernel functions can be chosen during the classification process namely; linear, polynomial, and Radial Basis Function (RBF) [50]. The kernels defined are:

- Linear: $K(x_i, x_j) = x_i.x_j$
- Polynomial: $K(x_i, x_j) = (x_i.x_j+1)^d$, where d is the degree of the polynomial kernel.
- quadratic kernel function: $K(x_i, x_j) = (x_i.x_j + 1)^2$
- Radial basis function: $K(x_i, x_j) = exp[-\gamma\,||x_i - x_j||^2]$

4.4.3 K Nearest Neighbour

The KNN classifier is a supervised machine learning technique which works well on many classification problems. The KNN has been widely applied in a large

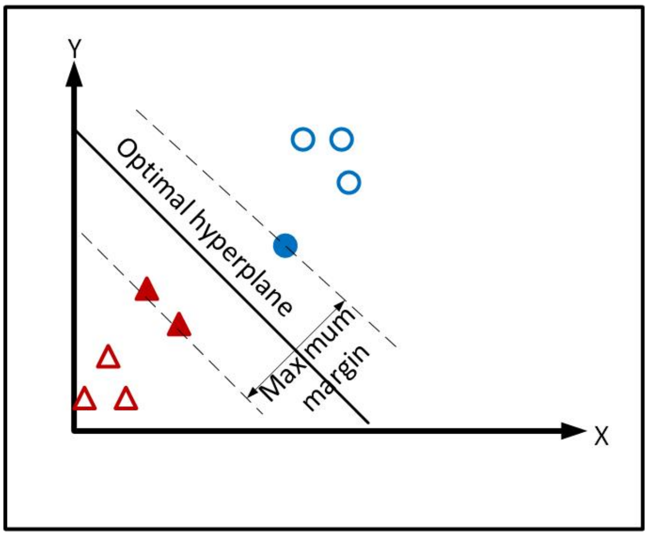

Figure 4.4: SVM maximum margin hyperplane [4].

number of classification and regression problems, including human behaviour recognition and fall detection [57]. KNN as a non-parametric method, is useful in classification tasks, where the decision boundary is very irregular. The main idea of the KNN method is to find a predefined number of training samples closest in distance to the classified example, and then to predict the label from these samples. KNN simply uses the training data itself for the classification and requires no learning process [13]. Thus, the cost of the learning process is zero and all the cost is dedicated to determining the decision. The decision is the most common label among the K closest neighbouring points. The parameters of the algorithm are the number of K neighbours and the procedure for combining the predictions of the K examples. Changing K can alter the decision of the classifier [63]. Various voting strategies in the literature are: Standard voting, Weighted voting, and Distance-based voting. The main concept of the KNN classifier is shown in Figure 4.5.

Given a test sequence x, its k closest neighbors $y_i...y_k$ are found and voting is conducted to assign the dominant class to x. This class of x is denoted by $c(x)$, and determined by the following equation:

$$C(x) = argmax_{c \in C} \sum_{i=1}^{k} \delta(C(y_i), C(x_i)). \tag{4.35}$$

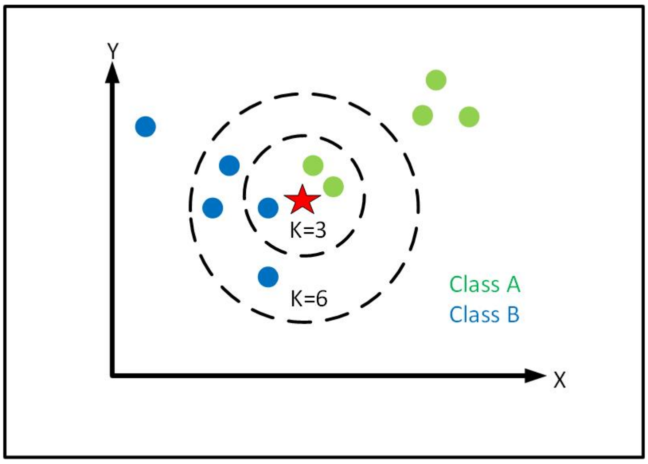

Figure 4.5: The main concept of KNN [2] .

where $C(y_i)$ is the class of (y_i) [124].

4.4.4 Bagging

Bootstrap aggregation also known as (bagging) method is proposed by Breiman [21] in order to improve the precision of classification results. Bagging method is widely applied to reduce the variance and increase the prediction accuracy of a learning algorithm by combining together hundreds or thousands of trees into a single prediction model. This method takes training subsets from the training data and produce a separate prediction model for each training subset and average the resulting predictions as illustrated in Figure 4.6. Bagging can be used for classification and regression problems.

The base version of bagging strategy performs experiments over various samples of the training data set. A classifier is generated for each of the training samples by a selected machine learning algorithm. Thus, for N of training samples, there are N particular classifiers. The result will be given as a combination of individual particular classifiers. There are other bagging strategies including:

- bagging like strategies- In this strategy, all the training sets are split into N subsets of the same size and each subset is used to create one classifier. A compound classifier is created as the aggregation of particular classifiers.

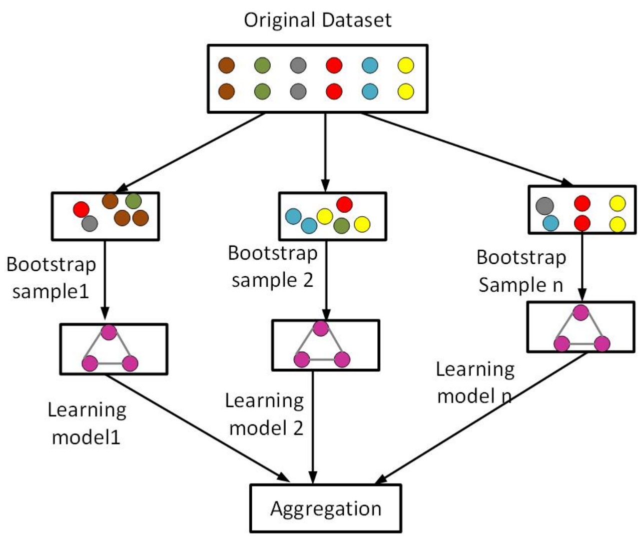

Figure 4.6: Bootstapping and learning ensembles [1].

- the disjoint partitions strategy- In this strategy, if N subsets are selected from the original training set, then each of them contains 1/N part from the original set. For large training sets, partitions enable parallel learning of base classifiers. Classifiers learnt on disjoint partitions reach the best results from bagging like strategies.

- small bags strategy- In this strategy, each subset is generated independently from the other subsets by random selection of training samples with the possibility to select a subset repeatedly. A subset can be located in several subsets. A combined classifier is obtained from the aggregation of particular classifiers [72].

In the bagging method, individual decision trees are grown in deep. These trees will have both high variance and low bias. The only parameters required are the number of samples and the number of trees. A large number of models may take a long time to prepare, but will not over-fit the training data [71].

4.4.5 Cross Validation

In the learning process, the data is split into two subsets: training and testing data. The data is normally split into 70% for training data and 30% for testing.

The learning algorithms such as artificial neural network are usually have an over fitting problem. The over fitting problem occurs when the learning algorithm is well trained on the training data and is poorly performed on the testing data. Using cross validation method, the data is divided into three sets: training, validation and test sets; training data used to train and learn a model, validation set used to validate the model, test data used to estimate the error rate of the trained model.

There are various methods used for cross validation. Commonly used methods are listed below.

- Holdout cross validation: the test data is heldout during the training phase, thus, there is no overlapping between the training and testing datasets. One of the main drawback of this method is that all the validation data is not used during the training phase and the system performance is dependent on the choice of the training and testing subsets. Also, the data in the testing set may be significant for training and if it holdouts, the performance prediction becomes poor.

- K-Fold cross validation: the dataset is divided into k equally sized subsets. The training and validation are performed in k iterations. In each iteration, a model is trained on all subsets except one. The left out subset is used to test the model. The advantage of K-Fold cross validation is that all data samples are used for both training and testing.

- Leave one-out cross validation: all data samples except one observation are used for training and one instance of data is used for testing. It is a special case of K-fold cross validation.

- Repeated K-fold cross validation: the K-fold cross validation is executed many times [73].

4.5 Discussion

In this chapter, various techniques for detecting moving object from image sequences are discussed. The implementation of methods to extract features from the foreground object are also explained. Some machine learning techniques utilised for fall detection are reviewed in this chapter. In Chapter 6 and 7 the extracted features along with machine learning techniques are used to implement algorithm for fall detection of elderly. Chapter 7, will describe the experimental results of applying machine learning techniques and a comparison between them. The techniques are evaluated using datasets that are mention in Chapter 3.

Chapter 5

Fall Detection Approach Using Threshold-Based Methods

5.1 Introduction

Visual-based monitoring systems are capable of providing information on falls and also other daily living behaviours. Falls can be detected by employing an automatic fall detection algorithm using intelligent surveillance systems [87]. Threshold-based methods are widely applied to detect falls by using manually pre-defined threshold values which do not require a learning step to classify falls [64]. In this study, threshold-based methods to detect falls in a home environment are proposed. The first approach combines motion information and changes in the orientation and the ratio of the ellipse to detect a fall event. The second approach employs three features; motion information, human shape variation and projection histogram to detect a fall.

This chapter is organised as follows: Section 5.2 presents the results of the background subtraction. Section 5.3 presents a fall detection approach based on a combination of motion information and change in the human shape. Section 5.4 presents a fall detection approach based on a combination of motion information, orientation and projection histogram. Finally, the discussion of the results is presented in Section 5.5.

5.2 Background Subtraction Algorithm

The first stage for the proposed fall detection approach is the detection of moving objects from a video sequence. This is performed via the background subtraction algorithm explained in Section 4.4.1.

In the segmentation algorithm, the foreground images are extracted from the background. The inputs to the segmentation algorithm are the background image and the current frame. The idea is that any pixel in the current frame can be part of the foreground if its value is different enough from its corresponding value in the background reference image. Firstly, the RGB image is converted into grey scale and then an absolute difference between the current frame and the background reference frame is computed for each pixel to produce the binary image.

Thereafter, a median filter and morphological operations are used to remove the noise from the binary image, fill holes and remove small components as well as improve the segmentation process. In our experiments, different window sizes for the median filter and different combination of dilatation and erosion operations were tested. The best results were obtained with a 5×5 median filter followed by dilation operation.

Figure 5.1 shows the segmentation algorithm results for a person walking in the room.

5.3 Fall Detection Based with Combination of Motion Information and Change in Human Shape

This section presents a novel approach for detecting falls based on a combination of motion information and human shape variation. The motion information of a segmented silhouette can provide a useful cue for classifying different behaviours. Also, the variation in human shape can be used to estimate the pose and hence fall events. The approach presented here extracts motion information, use the variation in shape and in addition uses the best-fit approximated ellipse around

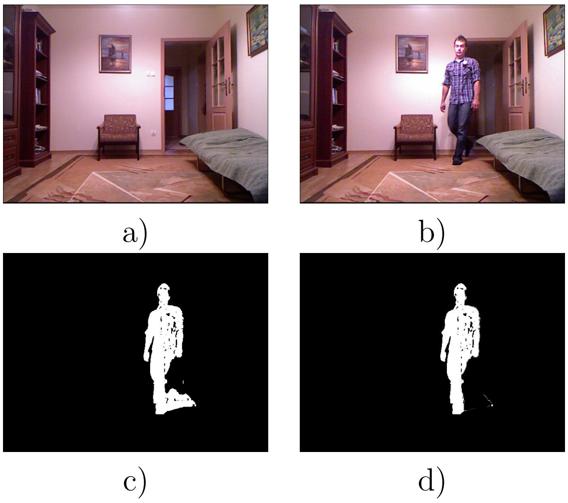

Figure 5.1: Background subtraction algorithm results: a) the background reference frame, (b) the current frame, (c) binary image (d) the improved binary image.

the human body to further improve the accuracy of fall detection. The proposed approach combines a motion feature and a change of the orientation and ratio of the ellipse features to detect a fall event.

An overview of the proposed system is presented in Section 5.3.1 and the results obtained from the experiments are shown in Section 5.3.2

5.3.1 System Overview

The proposed approach is developed for fall detection in a home environment and detecting a fall event based on motion information and changes in the orientation regarding the shape of a person. An overview of our fall detection system is shown in Figure 5.2. Firstly, background subtraction is implemented to segment out moving objects. Afterwards, useful features such as motion, ratio and shape orientation for detecting falls are extracted. To this aim, two features: motion and changes in shape are combined for detecting falls.

The first step of the system is to analyse the motion occurring in a given time window, using the tMHI [66]. The proposed approach is based on the assumption that the motion is large when a fall occurs. Therefore, the proposed system aims to detect a large motion on the person in the video sequence using tMHI. Then,

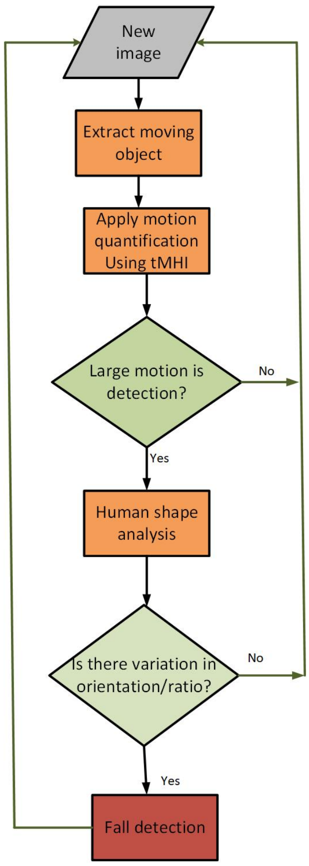

Figure 5.2: Flow diagram of the proposed method for human fall detection using motion, ratio and orientation.

we quantify the motion by calculating the pixel value of motion history image blob in the current frame, which is then divided by the total number of pixels in the human silhouette [116]. Figure 5.3 shows typical MHI images for turning around, walking and falling events.

A second step was taken to analyse the change of the human shape to identify a fall among other daily activities. An analysis of the moving object is performed by fitting an approximated ellipse around the human body. The orientation and the ratio of the fitted ellipse provide convenient information about the body posture [41]. Figure 5.4 shows the approximated ellipse around the human body and changes in the ellipse orientation.

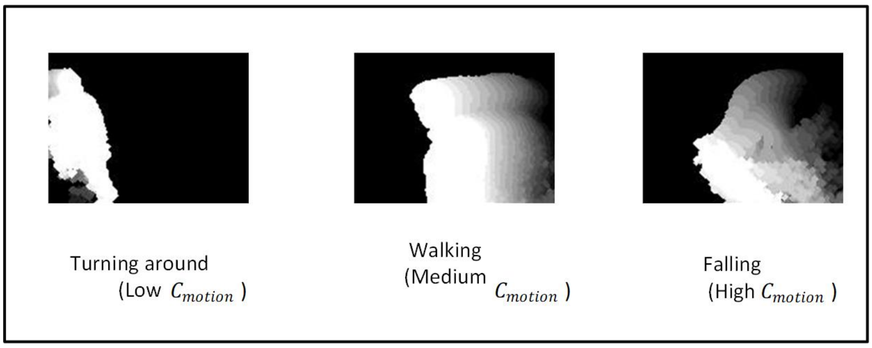

Figure 5.3: MHI images for some activities.

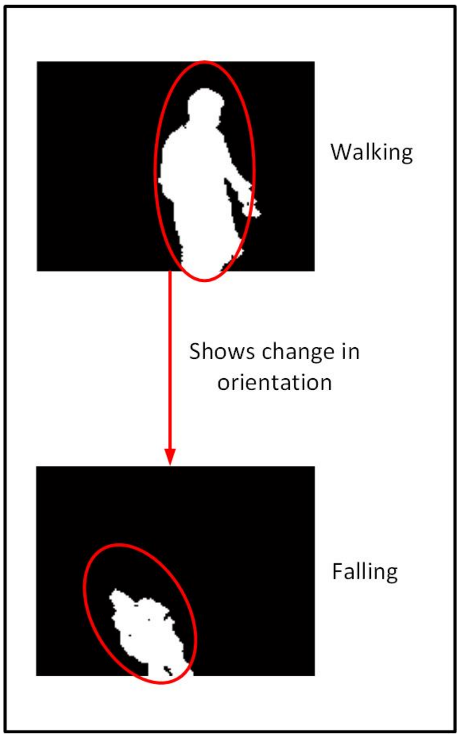

Figure 5.4: The change in orientation of human shape.

5.3.2 Combination of Motion Information and Change in Orientation

In order to evaluate the performance of the proposed fall detection approach, it is tested using the publicly available, Le2i fall detection dataset. Our proposed system considers an indoor environment with a stationary camera monitoring a single person. Table 5.1 illustrates the changes on the orientation and the ratio of ellipse during daily activities and simulated falls. For each activity, the orientation and ratio are computed for every frame. Based on previous study [43], the length of the sliding window is 2 seconds and the number of frames per window is 30 frames.

The size of the tMHI is 240 $\times$ 320, which is the same width as that of the frames in the videos. The tMHI is dependent on the duration, τ, and the decayr, δ. The duration indicates the temporal extent of the movement. Different duration values produce different tMHIs.

For each activity, we compute the orientation and ratio of the ellipse in each frame. Then, the standard deviation of the orientation and the ratio are computed using (4.30). The coefficient of motion values are used for fall detection. When a fall occurs, a large motion appears (high C_{motion}) with a significant change in orientation and ratio, as shown by their standard deviations in Table 5.2.

Based on our experiments, we consider that a large motion is a fall if:

Table 5.1: The change in orientation, θ, and ratio, ρ, of the approximated ellipse fitting for five different postures: walk, sit, bend,lie on the sofa and fall.

Activity	Orientation(θ)	Ratio(ρ)
Walk	1.527	3.576
Sit down	1.420	1.4756
Bend down	1.3407	2.5921
Lie on the sofa	1.1431	1.8622
Fall	0.5374	3.324

Table 5.2: Results of the combination of motion information (coefficients C_{motion}), and the change of human shape (standard deviation of orientation, σ_θ, and standard deviation of ratio, σ_ρ).

Activity	C_{motion}	σ_θ	σ_ρ
Walk	19.18	0.6831	0.4469
Sit	24.49	0.9394	0.0718
Bend	43.02	0.079932	0.2631
Lie	61.82	0.3608	0.2006
Fall	74.85	0.7460	0.4375

Figure 5.5: The values of the standard deviation of orientation and ratio of the fitted ellipse around the human body silhouette.

1. $C_{motion} > 65\%$,
2. the standard deviation of orientation, σ_θ, is higher than 0.60 and,
3. the standard deviation of ratio, σ_ρ, is higher than 0.35.

These thresholds were chosen manually based on observations of our video sequences. Figure 5.5 shows various activities with the corresponding values of the standard deviation of the orientation and the ratio.

For walk, bend and sit down activities, no large motion is detected and C_{motion} is lower than 65%. Therefore, the algorithm stops at the first step because of low motion. For lie activity, the motion is large and a possible fall is considered. However, the orientation standard deviation and the ratio standard deviation are below the fixed thresholds. Consequently, no fall is detected. The fall event was detected because the C_{motion} was higher than the selected threshold, $C_{motion} = 74.85$ and the standard deviation of the orientation, σ_θ, and the standard deviation of the ratio, σ_ρ, are higher than their corresponding thresholds. The values of the standard deviation of orientation and the standard deviation of ratio are 0.75 and 0.44 respectively.

In order to test the performance of tMHI against the original MHI, the same dataset is used to evaluate both algorithms. The MHI images and tMHI images are generated using (4.3) and (4.4) respectively.

The results show that different values of temporal duration, τ, provide different MHI. If the temporal duration value is smaller than the number of frames in a video sequence, then the motion information of the action is lost in the MHI image. If the current frame has no moving object and it is equal to the subsequent frame, then the output MHI is a black image . In our experiments the duration parameter τ is set to one and therefore, pixel values are reduced by one. In tMHI, the decay parameter is usually greater than one and in our experiments, the decay parameter, δ, in the tMHI is two.

To evaluate the results of our proposed system, we evaluate the coefficient of motion using MHI and tMHI methods as shown in (4.5) and (4.6). The comparison results for the two motion coefficients are shown in Table 5.3. It is clear from table 5.3 that tMHI detect fall event with motion variation higher than 65%. However, MHI fails to detect the fall as the motion coefficient is 64.59% and less than the proposed threshold.

5.4 Fall Detection Based on Combination of Motion, Orientation and Projection Histograms

This section presents a novel visual-based fall detection which employs three unique features; motion information, human shape variation and projection histogram to detect a fall. Motion information of a segmented silhouette, which when extracted can provide a useful cue for classifying different behaviours. Also, the projection histogram and variation in human shape can be used to describe human body postures and subsequently fall events. The proposed approach presented here extracts motion information, using best-fit approximated ellipse around the human body and in addition projection histogram features to further improve the

Table 5.3: Results of computing motion information (coefficients C_{motion}) using MHI and tMHI.

Activity	tMHI	MHI
Walk	3.9459	3.8995
Sit down	4.7019	4.5484
Bend down	25.6963	23.5094
Lie on the sofa	55.5288	55.5288
Fall	65.4999	64.5921

accuracy of fall detection. Experimental results are presented and show high fall detection rate of 99.81% with partially occluded video data.

An overview of the proposed system is presented in Section 5.4.1; and results obtained from our experiments are in Section 5.4.2

5.4.1 System Overview

This method was applied for monitoring human activities in a home environment and detecting a fall event based on motion information, changes in shape orientation and projection histograms. An overview of the proposed fall detection system is shown in Figure 5.6. Firstly, background subtraction is implemented to segment out moving objects [94]. Afterwards, useful features such as motion, shape orientation and histograms for detecting fall from different daily activities are extracted.

The first step is to analyse the motion occurring in a given time window, using the tMHI [66]. The proposed method is based on the assumption that the motion is large when a fall occurs. Therefore, the proposed system aims to detect a large motion of the person in the video sequence using tMHI. The motion is quantified by calculating the pixel value of motion history image blob in the current frame, which is then divided by the number of pixels in the human blob. A large motion does not necessarily signify a fall, as activities like fast walking or running may also exhibit such characteristics [95]. Considering that the first step is not sufficient, a second step was taken to analyse the change of the human shape to identify a fall among other activities. Analysis of the moving object is performed by fitting an approximated ellipse around the human body. The orientation of the fitted ellipse provides convenient information about the body posture [41].

It is assumed that human fall have greater acceleration than other daily activities. However, focusing only on a fast acceleration can result in many false alarms during fall-like activities like sitting down quickly [61]. Therefore, combining motion and orientation features helps to discriminate actual fall from other activities. After the ellipse fitting, the orientation of the ellipse, θ, is taken as a feature to describe a human body posture. The motion feature, C_{motion}, indi-

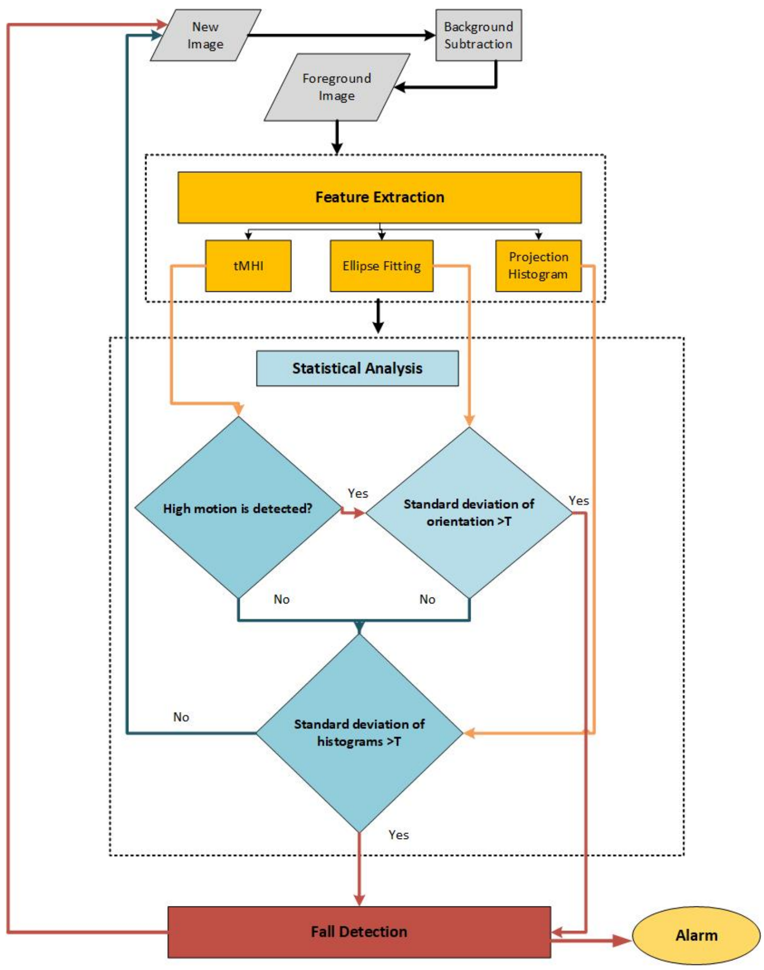

Figure 5.6: Flow diagram of the proposed method for human fall detection.

cates the changing rate of human motion and standard deviation of orientation indicates the changing rate of human shape [9].

Common fall detection algorithms are based on the fact that the fall activity have high acceleration than other normal activities, this study highlights the situation when human fall can happen with low motion rate. A typical example is when a person loses balance and hold onto a furniture to prevent a fall. Therefore, the third feature which is projection histogram feature is applied to confirm a fall event.

5.4.2 Combination of Motion, Orientation and Projection Histograms

To evaluate the robustness of the fall detection algorithm described in this work, video data recorded over a period of time and described in Section 4.4.1 was used. The recorded video is manually segmented into short video clips containing some activities like walking, setting, bending and fall. In order to segment the foreground object from the background, a simple background subtraction algorithm [94] is performed to extract the silhouette from the background. After the silhouettes are acquired through segmentation, features like motion, orientation of ellipse around the human silhouette and projection histograms are extracted.

The tMHI is then computed to quantify the motion in each time window. In addition, shape changes of the extracted silhouettes are analysed and difference of horizontal and the vertical histograms computed. tMHI is dependent on two parameters; the duration, τ, and the decay, δ, parameters. The duration parameter decides the temporal extent of the movement and different duration values produce different tMHIs. When a fall occurs, a large motion appears (high C_{motion}) with a significant change in orientation. In this work, the period of the sliding window is set to 2 seconds with 30 frames per window. For each activity, the orientation of the silhouette in each frame and the standard deviation of orientation are computed using (4.30). Then the standard deviation of the difference between horizontal and vertical histograms are calculated.

Since the proposed fall detection algorithm is designed mainly for low-cost low-resolution RGB cameras, recorded data from Cameras #3 (as described in Section 4.4.1) is used for testing. In our experiments a large motion is described as a fall if all of the following three conditions are met:

1. if $C_{motion} > 30\%$,
2. if the standard deviation of orientation σ_θ is higher than 0.40,
3. if the standard deviation of difference between horizontal and vertical histogram is higher than 20.

The threshold values used in the three conditions were empirically chosen based on observation of our video sequences. When the standard deviation of the difference between the horizontal histogram and vertical histogram is increased dramatically, a fall is considered to have occurred.

The results of combining motion information, human shape analysis and the projection histogram feature from Camera #3 are as shown in Table 5.4. For walking, bending down, lying and sitting down activities, there is no large motion associated with them and their computed C_{motion} is lower than 30%. In addition, the orientation standard deviation for these activities are below the fixed thresholds, therefore, a fall event won't be detected. A fall event would be detected because the C_{motion} is higher than the selected threshold, $C_{motion} = 41.20\%$ and the standard deviation of orientation σ_θ higher than the determined thresholds 0.44.

Examples of normal activities like bending and lying, and fall events are shown in Figure 5.7. For each image the corresponding tMHI image as well as the approximately fitted ellipses are shown.

In order to test the robustness of the proposed fall detection algorithm, data recorded with different cameras mounted at different angles (ceiling-mounted) to the prime test camera have also been used. Two ceiling-mounted cameras were used for the robustness test and have different threshold values compared to the tripod-RGB camera, for detecting fall events. In our experiments a large motion is described as a fall if all of the following three conditions are met:

1. if $C_{motion} > 30\%$,

2. if the standard deviation of orientation σ_θ is higher than 1.60,

Table 5.4: Combined results of motion information (coefficients C_{motion}), standard deviation of orientation, σ_θ, and standard deviation of the difference between horizontal and vertical projection histogram, $\sigma_{(H-V)}$, using Camera #3.

Activity	$C_{motion}(\%)$	σ_θ	$\sigma_{(H-V)}$
Walking	5.52	0.01	2.78
Sitting	12.55	0.07	7.33
Bending	4.04	0.03	5.48
Lying	9.94	0.05	3.20
Fall	41.20	0.44	42.58

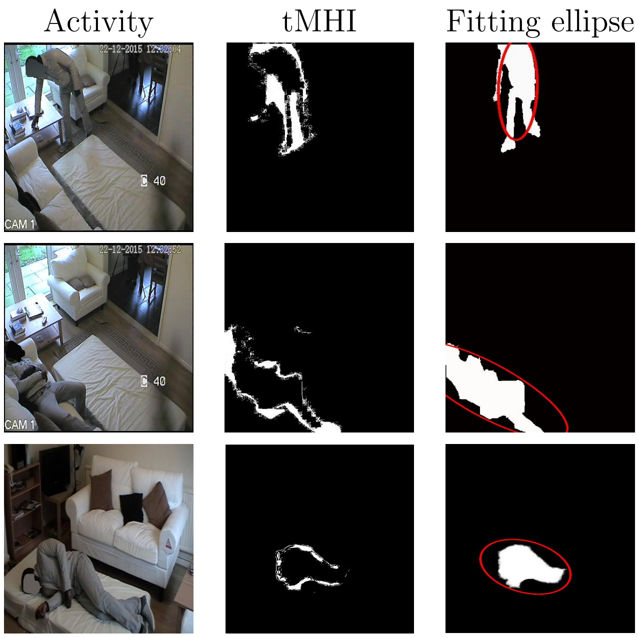

Figure 5.7: The tMHI feature and fitting ellipse around the human body for the human actions.

3. if the standard deviation of difference between horizontal and vertical histogram is higher than 20.

These thresholds were empirically chosen based on observation of the ceiling-mounted video sequences. The results of combining motion information, human shape analysis and projection histogram features using the front mounted ceiling Camera #2 are shown in Table 5.5.

A fall is detected in Table 5.5 because the C_{motion} is higher than the selected

Table 5.5: Results of combination of motion information (coefficients C_{motion}), standard deviation of orientation, σ_θ, and standard deviation of the difference between horizontal and vertical projection histogram, $\sigma_{(H-V)}$, for Camera #2.

Activity	$C_{motion}(\%)$	σ_θ	$\sigma_{(H-V)}$
Walking	29.50	1.59	10.17
Sitting	31.84	1.61	14.70
Bending	19.36	0.71	8.87
Laying	34.74	0.29	11.09
Fall	33.61	1.64	34.14

threshold, $C_{motion} = 33.61\%$ and the standard deviation of orientation, σ_θ, is higher than the selected threshold 1.64.

The proposed algorithm starts by testing the motion for different activities. For lying and sitting, the motion element is large and a possible fall is considered. Also, the standard deviation of the orientation is higher than the fixed thresholds when the person is sitting. This is mainly due to occlusion of the foreground silhouette. Relying solely on the two features (motion and orientation's standard deviation) would classify the sitting event as a fall. Therefore, other local features such as projection histograms are applied as described in Section 4.3.3 to correctly classify the sitting event.

As can be seen from Table 5.5 the $\sigma_{(H-V)}$ values of the none-fall activities are below the selected threshold. The results show that the extra feature can effectively distinguish fall among other activities. According to the results, using just motion history and analysis of the human shape features, would result in some sequences like sitting and lying incorrectly classified as fall events. However, the combination of motion history, change in shape orientation and change in horizontal and vertical projection histograms eliminates this false detection.

Similarly, the proposed fall detection algorithm was tested using video data from side-facing ceiling mounted Camera #1. The results of combining motion information, projection histograms and human shape analysis from video data using Camera #1 are shown in Table 5.6. Just like the previous two camera data, the first step was to extract the motion features for the different activities. For bending, the computed motion feature is greater than the selected threshold 33.25%, but the σ_θ and $\sigma_{(H-V)}$ values are below the selected thresholds. In contrast, the actual fall event has low motion feature of 23.81% compared to a threshold value of 30% while the σ_θ and $\sigma_{(H-V)}$ are greater than the corresponding thresholds. In this case the three conditions were not met, therefore, the fall event has been misclassified.

The proposed algorithm has further been tested using the publicly available Le2i fall detection dataset, also been used in [113] and [48]. The results of the proposed fall detection algorithm using the Le2i fall detection dataset are presented in Table 5.7.

From Table 5.7 the lying and sitting activities have associated motion feature greater than the threshold, as if they are fall events. However, the standard deviation of the orientation is lower than the fixed thresholds. Also, the projection histograms for these activities have a small value. The fall event is detected because the motion and the standard deviation of the orientation are higher than threshold values. In addition, there is a big change in the standard deviation of the difference between horizontal and vertical histograms when the person falls down.

5.4.3 Performance Evaluation

In order to evaluate the performance of the proposed algorithm, each video data was manually segmented into a number of video segments. Each video segment contains one of the considered activities and its manually labelled as fall or non-fall segments. As can be observe from Table 5.8, there are 21 video segments containing fall events (positive samples) and 133 video segments containing other

Table 5.6: Results of combination of motion information (coefficients C_{motion}), standard deviation of orientation, σ_θ, and standard deviation of the difference between horizontal and vertical projection histogram, $\sigma_{(H-V)}$, for Camera #1.

Activity	$C_{motion}(\%)$	σ_θ	$\sigma_{(H-V)}$
Walking	15.80	0.81	11.34
Sitting	14.89	1.74	14.46
Bending	33.25	0.60	5.94
Lying	13.58	0.21	21.75
Fall	23.81	1.67	30.51

Table 5.7: Results of combination of motion information (coefficients C_{motion}), standard deviation of orientation, σ_θ, and standard deviation of the difference between horizontal and vertical projection histogram, $\sigma_{(H-V)}$, using Le2i fall detection dataset .

Activity	$C_{motion}(\%)$	σ_θ	$\sigma_{(H-V)}$
Walking	20.75	0.03	10.57
Sitting	58.19	0.50	2.08
Bending	29.12	1.34	5.76
Lying	49.96	0.06	2.75
Fall	67.58	1.87	20.45

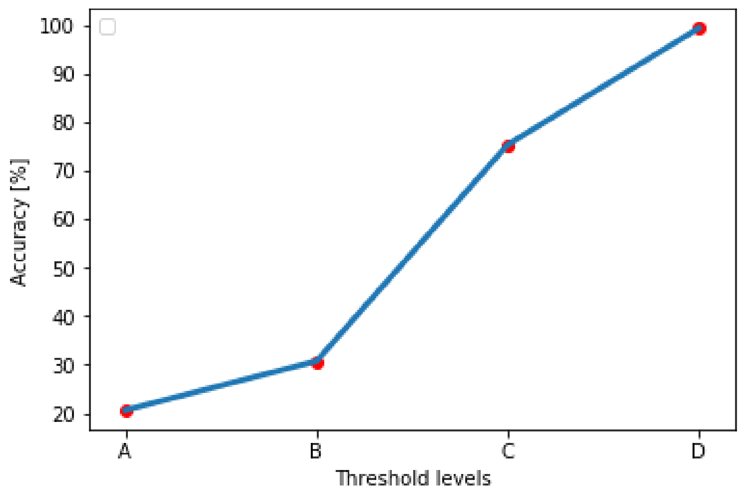

Figure 5.8: The accuracy results depend on threshold levels.

normal activities (negative samples).

In order to optimise the detection rate of the proposed fall detection approach, various threshold values of all features were tested as below.

- Case A: ($C_{motion} < 10$, $\sigma_\theta < 0.05$ and $\sigma_{(H-V)} < 5$).
- Case B: ($C_{motion} < 20$, $\sigma_\theta < 0.15$ and $\sigma_{(H-V)} < 10$).
- Case C: ($C_{motion} < 30$, $\sigma_\theta < 0.30$ and $\sigma_{(H-V)} < 15$).
- Case D: ($C_{motion} > 30$, $\sigma_\theta > 0.40$ and $\sigma_{(H-V)} > 20$).

Figure 5.8 presents the accuracy results of different threshold levels. Experiments results shown that the proposed fall detection approach achieves high accuracy of 99.82% by setting three conditions as specified in Case D.

The first row in Table 5.8, shows that the proposed system successfully detects human fall in 20 video sequence out of 21 videos and fails for one video sequence. As can be seen from the second row, all normal activities have been detected correctly as non-fall events. The performance evaluation is computed automatically given the labeled images and results from implemented fall detection algorithm.

There are four possible outcomes for testing a sequence as a fall event which are defined as follows:

- True Positive (TP): a video segment contains a fall, and is correctly detected as a fall.
- False Positive (FP): a video segment does not contain falls, but is incorrectly detected as a fall.
- True Negative(TN): a video segment does not contain falls, and is correctly detected as non-fall.
- False Negative(FN): a video segment contains a fall, but is incorrectly detected as not a fall.

The performance of the fall detector was evaluated with respect to accuracy, False Positive Rate (FPR), positive predictive value (precision), negative predictive value, sensitivity and specificity. They were calculated based on the definitions presented below.

- Accuracy=(TP+TN)/(TP+TN+FP+FN).
- The false positive rate= FP/(FP+TN).
- Positive predictive value (Precision)=TP/(TP+FP).
- Negative predictive value =TN/(TN+FN).
- Sensitivity= =TP/(TP+FN).
- Specificity= TN/(TN+FP).

Table 5.9 represents the performance of the proposed fall detection algorithm. The algorithm achieves 100% specificity and this means that all normal daily activities are assigned to the non-fall class. A perfect sensitivity implies that most falls are recognised as a fall event. The accuracy of human fall detection is 99.82% while maintaining small false alarms.

5.5 Discussion

In this chapter, efficient approaches for fall detection were proposed. The first approach is based on a combination of timed motion history and variation in human shape. The combination of motion and change in the human shape offers crucial information about human activities. Firstly, tMHI is implemented to quantify the motion of the person, then the person is approximated by an ellipse using moments to detect a change in the human shape by computing the orientation and ratio of the ellipse [87]. Moving objects is extracted using the tMHI method and then an analysis of the moving object is performed to detect changes in the human shape orientation and ratio.

The second approach is based on a combination of timed motion history, variation in human shape and projection histograms. The local feature which is based on projection histograms is applied to identify fall among other daily activity. A novel feature, namely the standard deviation of the difference between horizontal and vertical histograms is applied to confirm whether or not an event is a fall. Thus, when standard deviation of orientation is higher than the threshold value and a large motion information is detected.

Threshold-based fall detection methods are simple and cost- effective meth-

Table 5.8: Detection accuracy

Activity	Total Samples	Detected	Not detected
Falls	21	20	1
Activities of other daily living	133	133	0

Table 5.9: The performance of fall detection system based on motion, orientation and histogram features

Description	Obtained Value
Sensitivity	95.23%
specificity	100%
false positive rate	0%
Positive predictive value	100%
Negative predictive value	99.25%
Accuracy	99.82%

ods. However, the performance heavily depends on the fixed threshold level. Moreover, manually defining threshold values is difficult, as several activities of daily living (ADLs) like quickly sitting or laying can produce high motion which can be detected as fall and cause a high number of false alarms. Each dataset has different camera angle and lighting conditions. Thus, for each dataset it is necessary to determine threshold values that are the best to detect falls. Moreover, predetermine threshold values do not generalise well for unseen persons. Therefore, the system needs to be adapted for monitoring different people.

Chapter 6 and Chapter 7 will be conducted to apply various machine learning algorithms to automatically classify fall and non-fall activity, as the proposed fall detection algorithm is heavily reliant on pre-defined threshold values.

Chapter 6

Fall Detection Approach Using Neural Network

6.1 Introduction

This study presents a novel visual-based fall detection approach to support independent living for older adults through analysing the motion and shape of the human body. The proposed approach employs a new set of features to detect a fall. Motion information of a segmented silhouette can provide a useful cue for classifying different behaviours, while variation in shape and the projection histogram are used to describe human body postures and subsequent fall events. The proposed approach extracts motion information using the best-fit approximated ellipse and a bounding box around the human body produces projection histograms and determines the head position over time, to generate 10 features to identify falls. These features are fed into a multilayer perceptron neural network for falls detection.

The rest of this chapter is organised as follows: an overview of the proposed system is presented in Section 6.2. Section 6.3 describes in detail the features used. Section 6.4 discusses the use of a neural network for identification of falls. Details of the experiments conducted on the UR Fall Detection dataset is explain in Section 6.5. Section 6.6 presents the experiments performed on our recorded video dataset. Pertinent conclusions are presented in Section 6.7.

6.2 System Overview

This study proposes a method for monitoring human activities in a home environment and detecting a fall event based on motion information, changes in shape orientation, the position of the human head and projection histograms. An overview of the proposed fall detection system is shown in Figure 6.1. The proposed fall detection system includes four steps: data collection, foreground segmentation, feature extraction and fall detection. Background subtraction is implemented to segment out moving objects [94]. This method uses the difference between the current image and the background image to detect moving objects, which is in our case used to extract the human silhouette. The next step is to track the moving object to recognise its motion by identifying the object position in every frame. Afterwards, useful features such as motion information, shape orientation, temporal change of the head location and histograms for detecting a fall from other daily activities are extracted. The proposed method exploits motion, histogram and shape features based on the observation that human falls often involve drastic shape changes and abrupt motions as compared to other activities.

The first stage of the system is to analyse the motion occurring in a given time window, using tMHI [66]. The motion is quantified by calculating the pixel value of the motion history image blob in the current frame, which is then divided by the number of pixels in the human blob. The second stage is to analyse the change of the human shape. An analysis of the moving object is performed by fitting an approximate ellipse around the human body. The orientation of the fitted ellipse provides information about the body posture [41]. After the ellipse fitting, the orientation of the ellipse and the ratio between the major semi-axis a and the minor semi-axis b are used as features to describe human body posture in a general way. However, these features alone cannot describe postures in detail for distinguishing different activities [121], therefore further features were explained. A bounding box was used to surround the foreground object, then the y-coordinate of the top left point of the bounding box was computed and the absolute difference of y-coordinates in successive frames were used as features.

It is assumed that human falls have a higher acceleration than other daily

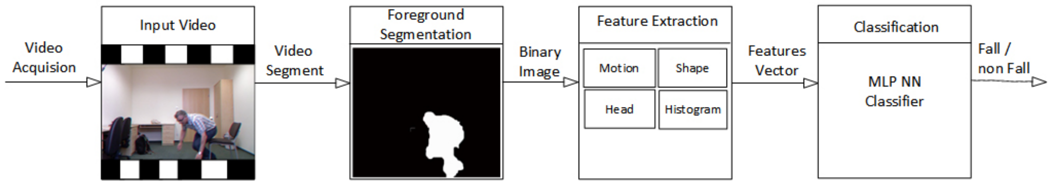

Figure 6.1: Flow diagram of the proposed human fall detection approach.

activities. However, focusing only on fast acceleration can result in many false alarms during fall-like activities like sitting down quickly [61]. Therefore combining motion with human shape helps to discriminate actual fall from other activities. After ellipse fitting, the orientation, ratio, major semi-axis and minor semi-axis are taken as features to describe human body posture. The motion feature C_{motion} indicates the changing rate of human motion and the orientation feature indicates the changes on the human shape [9].

Our previous study [10] highlighted the scenario of a human fall occurring at low speed. A typical example is when a person loses their balance, holds onto furniture to prevent a fall and yet still falls on the ground. Therefore an additional feature, the projection histogram, is computed to confirm a fall event [10]. Occlusion occurs when a relevant area of the person is covered or when the person moves behind an object and consequently part of his/her body is not visible by the camera [32]. To deal with the occlusion problem, the proposed fall detection approach tracks the human head in consecutive frames. Tracking the head position of the person can provide useful information, as the head tends to be visible most of the time. Finally, the extracted features are used as input vectors to the MLP neural network for falls and non-falls event classification.

6.3 Selected Features

Based on the information provided in Section 4.3, for each video sequence, the moving object is detected and 10 features. The selected feature vector F is given by:

$$F = [C_{motion}, \theta, \rho, a, b, Hz_{(y)} - Vt_{(x)}, y, \sigma_y, |y_i - y_{i-1}|, \sigma_{|y_i - y_{i-1}|}] \tag{6.1}$$

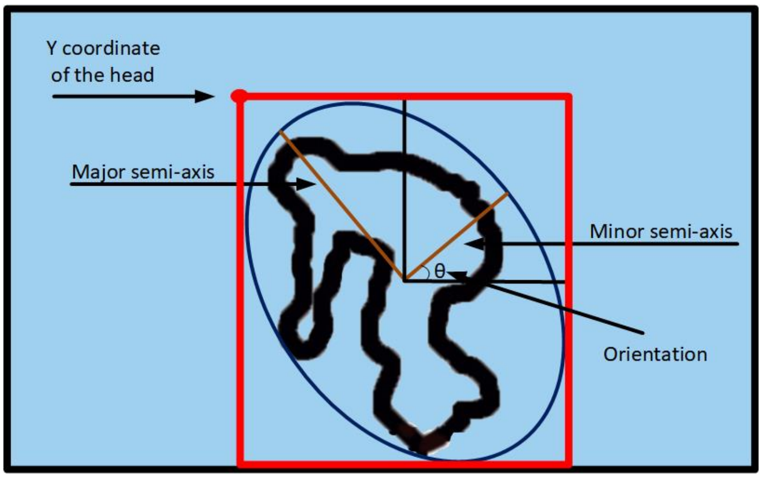

Figure 6.2: Ellipse fitting and bounding box around human body.

.

A description of the selected features is provided in Section 4.3.5. Figure 6.2 illustrates the features extracted from the fitted ellipse and the enclosed bounding box.

For every video sequence, 20 windows of frames with an overlap of 19 frames per window are used to extract features. For example, the first window contains frames from 1 to 20, the second window contains frames from 2 to 21 and so on. The features are extracted from every frame, and from every sequence of 20 frames, is generating the result feature vector which is then used as the input to the neural network to detect the fall. The features vector is then normalised.

6.4 Falls Detection with a Neural Network

In order to classify between falls and non-fall activities, the feature vector was fed as into one layer Multilayer Perceptron (MLP) Neural Network, which is shown schematically in Figure 6.4. The number of neurons in the hidden layer that gives higher accuracy is ten as shown in Figure 6.3. The hidden layer makes use of the scaled conjugate gradient algorithm. The use of this algorithm enables the network to perform well despite the dynamic range of the inputs, by reducing the number of iterations required when some features are much larger than others.

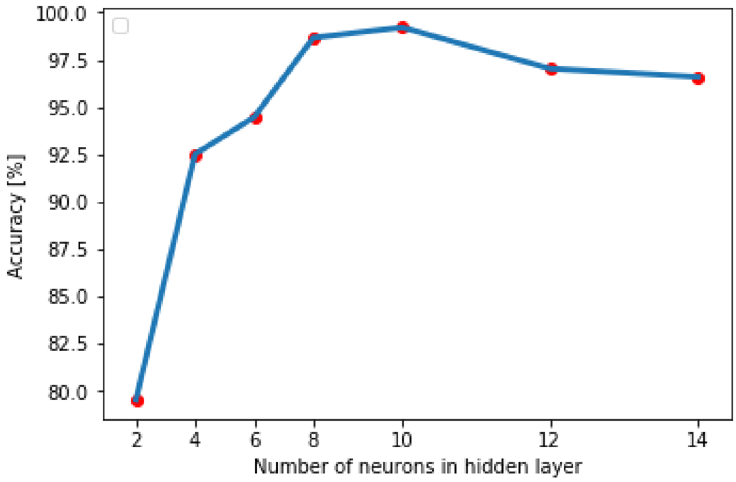

Figure 6.3: The accuracy of the proposed approach depend on the number of neurons in the hidden layer.

The outputs of the hidden layer are passed to the output layer, from where the decisive outputs are generated. A 10-fold cross-validation strategy was used to evaluate the methodology. The whole data was divided into 10 subsets of equal size, for each fold a NN model was trained on all the subsets except one . The left out subset was used to test the model. This process was repeated until all folds were used to either train or test the model. The algorithm for activities shown in Figure 6.6.

The data was previously normalised with zero mean and standard deviation equal to 1, so that it consisting with the transfer function was kept. A mean squared error function was chosen as the evaluation criterion. This function minimises the mean of the squares of the errors produced in each iteration and updates the network weights and biases accordingly.

The MATLAB neural network toolbox was utilised to build the ANN model. All features extracted from the images are gathered in a large cell array. Each row represents 10 features of an image. Figure 6.5 illustrate the process of detecting falls with a neural network.

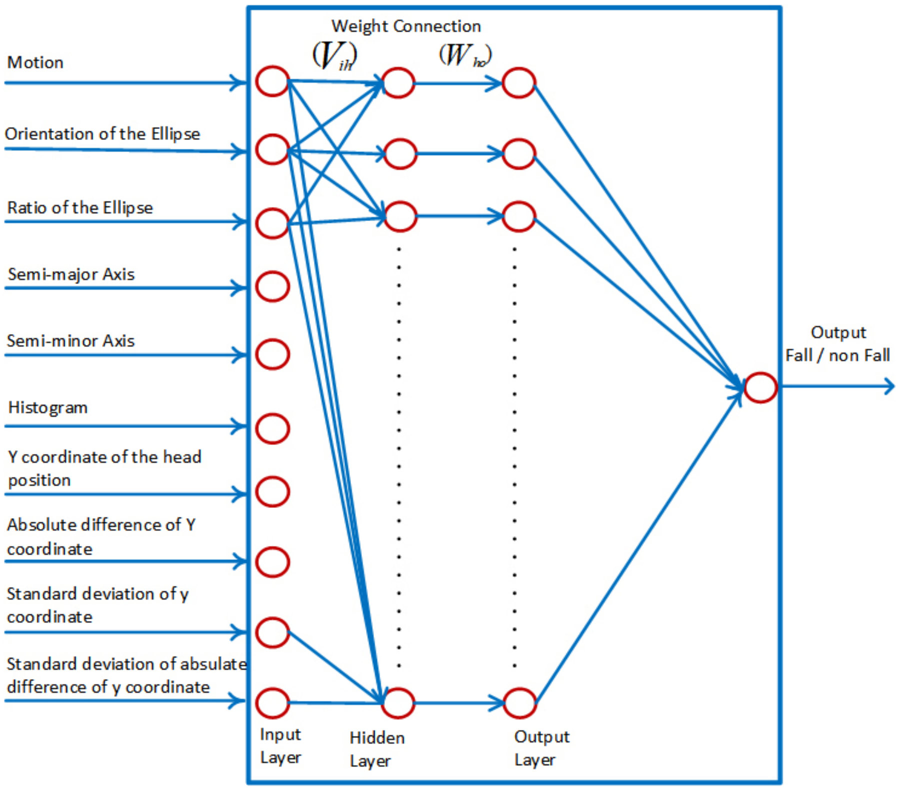

Figure 6.4: Fall detection neural network architecture.

6.5 Experiments with the UR Fall Detection Dataset

The first group of experiments were conducted using the UR Fall Detection dataset [62], which contains 30 fall scenarios recorded by two Kinect sensors and an accelerometer and 40 daily activities using one Kinect sensor parallel to the floor. The normal daily activities included walking, sitting, lying down, bending and crouching down. Some of these videos were recorded in low light conditions and some videos present examples of occlusion. Falls were simulated in different directions with respect to the camera view. Different types of fall incidents were recorded to include forward, backward and sideways falls.

In order to segment out the foreground object from the background, a simple background subtraction algorithm [94] was employed to extract the silhouette from the background. After the human silhouettes were extracted through segmentation, the second step was to extract useful features from the human sil-

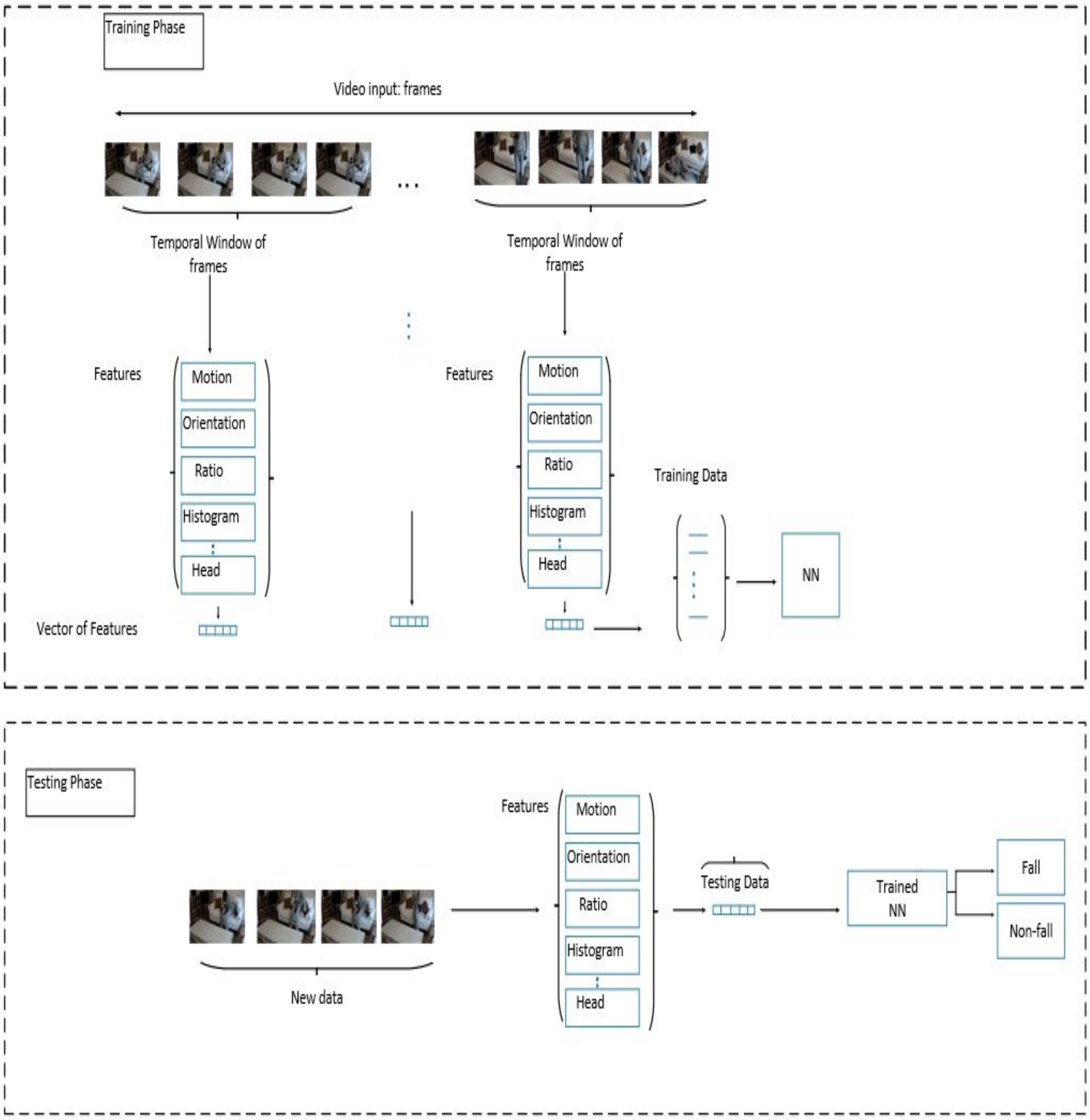

Figure 6.5: The procedure of the proposed approach to classify activities.

houette to detect falls. These are the features discussed and defined in Section 4.3. The values associated with each feature during a daily activity and a fall are shown in Figure 6.7.

The whole data samples (7177 samples with 10 features) were used.The experiments were done using 10-fold cross-validation. The training samples were created from the extracted feature vector and stored along with the target output values corresponding to the specific input patterns. The fully trained neural network was then used on the test data to classify the patterns associated with the activities of the humans and generate the outputs corresponding to the respective activities, which in our case were either fall or non-fall activities.

In order to optimise the final classification rate of the proposed approach, various combinations of features were tested. The features combinations are listed as below.

- Case A: The first row of the Table 6.1 presents the recognition results in which the input features of the MLP network used are motion, the orientation and the ratio of the ellipse, and the projection histogram features. This subset of features corresponds to those we used in our previous work [10] on our own dataset, and includes the main 4 motion features. The proposed projection histogram feature are computationally efficient and can effectively distinguish fall among other activities.

- Case B: The second row in Table 6.1 presents the recognition results where the input features of MLP network include motion, the orientation, the ratio, the major semi-axis and the minor semi-axis of the ellipse, and projection histogram. Thus this row adds the 2 semi-axes of the ellipse, giving more information on orientation, size and change of proportion (potentially showing someone crumpling as they fall).

- Case C: The third in Table 6.1 presents the recognition results obtained using features of motion, the orientation, the ratio of the ellipse, projection histogram, y-coordinate of the head point, the standard deviation of y-coordinate, the absolute difference of y-coordinate and the standard deviation of absolute difference of y-coordinate. The combine the basic motion characteristics with the detailed behaviour of the head in terms of its position and its position variance.

- Case D: In order to further validate our feature sets, the data in Figure 6.7 was examined, and only those features which exhibited obviously large changes were used. These were the motion, the y coordinate, the ratio and the major semi-axis (as shown in graphs a,c, e and g).

- Case E: The last row in the Table 6.1 shows the recognition results obtained using all features - motion, the orientation, the ratio, the major semi-axis and the minor semi-axis of the ellipse, projection histogram features, y-coordinate of head point, the standard deviation of y-coordinate, the absolute difference of y-coordinate and the standard deviation of absolute difference of y-coordinate.

6.5.1 Performance Evaluation and Recognition Results on the UR Fall Detection Dataset

The performance of the fall detector was evaluated with respect to accuracy, false positive rate, precision, sensitivity and specificity. They were calculated as stated in Section 5.4.3. In addition, two performance measures; the F-score and the False Negative Rate (FNR) were computed as follow:

- F-score= 2TP/ (2TP + FP + FN);

- The false negative rate= FN/(FN+TP);

Using all the features, our approach achieves 97.38% specificity, which means most daily activities are assigned to the non-fall class. A high sensitivity implies that most falls are recognised as a fall event. The accuracy of the human fall detection system is 99.24%. The proposed system shows precision of 99.60% while maintaining a low rate of false alarms.

From the curves in Figure 6.7, it is possible to infer that the more significant features characterising the fall events are: motion, ratio, the major-semi axis and the y-coordinate of the head point. Table 6.1 shows that these key features as in group D achieved similar accuracy to features group B which includes six features.

However, using only these features the system achieves an accuracy of about 92.13% and low specificity (true negative rate) 58.93%. This means that some normal activities such as sitting down and lying are detected as falls, and these features are not sufficient to discriminate a real fall from a person lying or sitting down. It appears that these features cannot always distinguish between activities when there is a high degree of similarity in terms of high motion, or when the ratio of the approximated ellipse is nearly the same for both non-fall and fall activities. This can give rise to false positives.

The performance values (sensitivity of 99.52%, specificity of 97.38%, precision of 99.60% and accuracy of 99.24%) seem to be good considering that we are using only RGB images and modest hardware.

The experiments were repeated using ANN with two hidden layers; each hidden layer has 10 neurons. The whole data samples (7177 samples with 10 features) were used with 10-fold cross-validation method. Table 6.2 shows the recognition results of the proposed fall detection approach using two hidden layers. The proposed approach shows a high detection rate of 99.39% while when using only one hidden layer the detection rate was 99.24%. However, when using one hidden layer the Epoch= 6 iterations while with two hidden layer the Epoch=22 iterations. Therefore, using ANN with one hidden layer is more effective for the purpose of fall detection.

6.6 Experiments on Recorded Video Data

The second group of experiments were performed on the collected video dataset. The whole database contains 23954 samples with 10 features. A 10-fold cross-validation strategy was used to evaluate the methodology.

Table 6.1: Recognition results on UR Fall Detection datasets using different combination of features.

Method	Accuracy	F-Score	FPR	FNR	Precision	Sen.	Spe.
Case A 4 features	88.69%	93.82%	79.02%	0.91%	89.08%	99.09%	20.98%
Case B 6 features	92.77%	95.94%	44.75%	1.46%	93.47%	98.54%	55.25%
Case C 8 features	96.35%	97.91%	16.89%	1.61%	97.43%	98.39%	83.11%
Case D 4 key features	92.13%	95.54%	41.07%	0.027%	93.91%	97.23%	58.93%
Case E 10 features	99.24%	99.56%	02.62%	0.48%	99.60%	99.52%	97.38%

Again, in order to evaluate the performance of the fall detection approach, the same subsets of input features.

- Case A: The first row in Table 6.3 presents the recognition results in which the input features of MLP network used are motion, the orientation and ratio of the ellipse, and projection histogram features.

- Case B: The second row in Table 6.3 presents the recognition results where the input features of MLP network include motion, the orientation, the ratio, the major semi-axis and the minor semi-axis of the ellipse, and projection histogram features.

- Case C: The third row in Table 6.3 presents the recognition results obtained using features of motion, the orientation, the ratio of the ellipse, projection histogram, y-coordinate of the head point, the standard deviation of y-coordinate, the absolute difference of y-coordinate and the standard deviation of absolute difference of y-coordinate.

- Case D: The last row in Table 6.3 shows the recognition results obtained using all features - motion, the orientation, the ratio, the major semi-axis and the minor semi-axis of the ellipse, projection histogram features, y-coordinate of head point, the standard deviation of y-coordinate, the absolute difference of y-coordinate and the standard deviation of absolute difference of y-coordinate.

Table 6.2: Recognition results on UR Fall Detection datasets using two hidden layers.

Description	Performance Value
Accuracy	99.39%
F-Score	99.65%
Precision	99.45%
Sensitivity	99.85%
specificity	96.40%

6.6.1 Performance Evaluation and Recognition Results on the Recorded Video Dataset

The experiments were performed using 10 fold cross-validation. The results using different combination of features are given in Table 6.3. The Experiments results show that the highest recognition accuracy (99.94%) was obtained using all the proposed features.

The features extracted from the head of the person (in Case C), improve the performance of the proposed approach, by using the features extracted from the head the accuracy improved from 98.95% to 99.89%, the specificity increased from 64.19% to 97.82%. From the results reported in this chapter, it can be concluded that every feature in the proposed approach is essential to accurately identify falls.

Using all the features, the algorithm achieves 98.69% specificity and this means that most daily activities are assigned to the non-fall class. A high sensitivity implies that most falls are recognised as a fall event. The proposed system shows a high detection rate of 99.94% while maintaining a low rate of false alarms.

6.6.2 Evaluating the Proposed Fall Detection in Different Environments

In order to evaluate the proposed fall detection approach, various experiments were carried out in different environments performed by different participants. These experiments are developed as described below.

Table 6.3: Recognition results on our recorded video dataset using different combination of features.

Method	Accuracy	F-Score	FPR	FNR	Presision	Sen.	Spe.
Case A 4 features	98.95%	99.46%	0.358%	3.183%	98.97%	99.97%	64.19%
Case B 6 features	99.06%	99.52%	0.327%	1.273%	99.05%	99.99%	67.25%
Case C 8 features	99.89%	99.95%	0.021%	4.456%	99.94%	99.96%	97.82%
Case D 10 features	99.94%	99.97%	0.013%	2.546%	99.96%	99.97%	98.69%

- Case A: The first row in Table 6.4 presents the recognition results in which the fall detection approach employs neural network that was trained and tested using the same person (A) in the same environment (Living room).

- Case B: The second row in Table 6.4 presents the recognition results where the neural network was trained and tested by the same person in different environments. Thus, for example, the neural network was trained using person (A) in environment (Living room) and tested by using the same person (A) in different environment (Bedroom).

- Case C: The third row in Table 6.4 shows the recognition results of the neural network trained using different persons in the same environment. That is, the neural network was trained using person (A) in environment (Living room) and tested on person (B) in the same environment (Living room).

- Case D: The last row in Table 6.4 shows the recognition results of the neural network using different persons in different environments. That is, the neural network was trained using person (A) in environment (Living room) and tested on person (B) in different environment (Bedroom).

The experiments were performed using all ten features, with ten input neural to the neural network, The 10 K-fold cross validation method was used. In all cases, both training data and testing data contain various ADL events and falls in various directions.

The proposed approach achieves high detection rate when presented with a new, unfamiliar domestic environment. The results shown in Table 6.4 show that the proposed approach is able to perform well in different scenarios. In all cases the proposed approach obtains outstanding performance in terms of accuracy (100%), sensitivity (100%) and specificity (100%).

6.7 Discussion

The work presented in this chapter has focused on investigating a relatively low-cost and reliable fall detection approach for older adults based on computer vision

Table 6.4: Recognition results of the proposed fall detection using neural network.

Method	Case A	Case B	Case C	Case D
Accuracy	100%	100%	99.54%	100%
F-Score	100%	100%	99.96%	100%
FPR	0%	0%	0.01%	0%
FNR	0%	0%	2.35%	0%
Precision	100%	100%	99.95%	100%
NPV	100%	100%	99.47%	100%
Sen.	100%	100%	99.98%	100%
Spe.	100%	100%	98.94%	100%

techniques. The approach presented here employs enhanced features which are extracted from the human silhouette. These features are the motion information, orientation, ratio, the major semi-axis and the minor semi-axis of the fitting ellipse, the projection histogram, the y-coordinate of the head point, the standard deviation of y-coordinate, the absolute difference of y-coordinate and the standard deviation of absolute difference of y-coordinate.

Experimental results show that the proposed algorithm is reliable for fall detection. The proposed approach is based on a combination of timed motion history and variation in human shape. The combination of motion and change in the human shape offers crucial information about human activities. Firstly, tMHI is implemented to quantify the motion of the person. The person is approximated by an ellipse using moments to detect a change in the human shape by computing the orientation, the ratio, the major semi-axis and the minor semi-axis of the ellipse [87]. In addition, the local feature based on projection histograms is applied to identify a fall among other daily activities. Tracking of head position improve the performance of the presented system. These features were fed into a MLP Neural Network for fall classification. It can be observed that the proposed algorithm produces a high recognition rate of 99.60% while maintaining a low false alarm rate of 2.62%.

The combination of features is significant. The histogram feature helps in identification of sudden changes in the human body shape. Adding the orientation feature helps in characterising a human's fall because it has less variation during the frames of the video sequence until a fall happens. Then the orientation will change suddenly until the human body reaches a prone position after the fall.

When a large motion is detected, the orientation and ratio of the ellipse change suddenly and at the same time, the histograms decrease rapidly. This means a fall event is considered possible. The results have shown that the use of major and minor semi-axis of the ellipse contributes towards a more accurate fall detection system. Additionally, head features are applied for reliable classification of motion and determination of a fall event.

From the analysis, the system achieved the best performance and lowest false alarm rate after adding the difference between the horizontal and vertical projection histograms, the standard deviation of y-coordinate and the absolute difference of y-coordinate as input features. The detection results obtained using head features have shown that the best performance regarding accuracy and specificity is obtained when using all ten features together.

The experiments performed on the publicly available fall detection datases; UR Fall Detection dataset were used to verify the universality of the proposed approach.

In Chapter 7, various machine learning methods including SVM, KNN and bagging tree are tested and compared against the performance achieved by the use of the MLP.

Algorithm 1 **Fall Classification**

Require: Input a sequence of video frames; 1, 2, 3, ..., N

/* Extracting moving object*/

1. for i=1: N do
2. Segment moving object using background substation algorithm.
3. denoising by median filter.
4. Save binary images.

/* Extracting features */

5. j = 1
6. While j ≤ N do
7. /* Extracting motion feature */
8. Compute coefficient of motion C_{motion}
9. /* Extracting shape features from best fit ellipse*/
10. Compute major semi-axis a, minor semi-axis b, ratio ρ, orientation θ.
11. /* Extracting projection histogram feature*/
12. $Hz_{(y)} - Vt_x$
13. /* Extracting head features*/
14. $y - coordinate$ of head point (y)
15. The standard deviation of $y - coordinate$ (σ_y)
16. The absolute difference of $y - coordinate$($|y_i - y_{i-1}|$)
17. The standard deviation of the absolute difference of $y - coordinate$ ($\sigma_{|y_i - y_{i-1}|}$)
18. /* Insert all the extracted features into a matrix Fs */
19. $Fs = \{C_{motion},\ a, b, \rho, \theta, Hz_{(y)} - Vt_x, y, \sigma_y, |y_i - y_{i-1}|, \sigma_{|y_i - y_{i-1}|}\}$

/*Training Machine Learning Classifier */

20. NormalisedData= Normalise (Matrix Fs)
21. (TrainingData, TestingData) = Split (NormalisedData)
22. ClassiferModel= Train (TrainingData, TargetRating)
23. sequence labelling= Test (ClassifierModel, TestingData)
24. Output: Fall or non-fall

Figure 6.6: Fall classification algorithm.

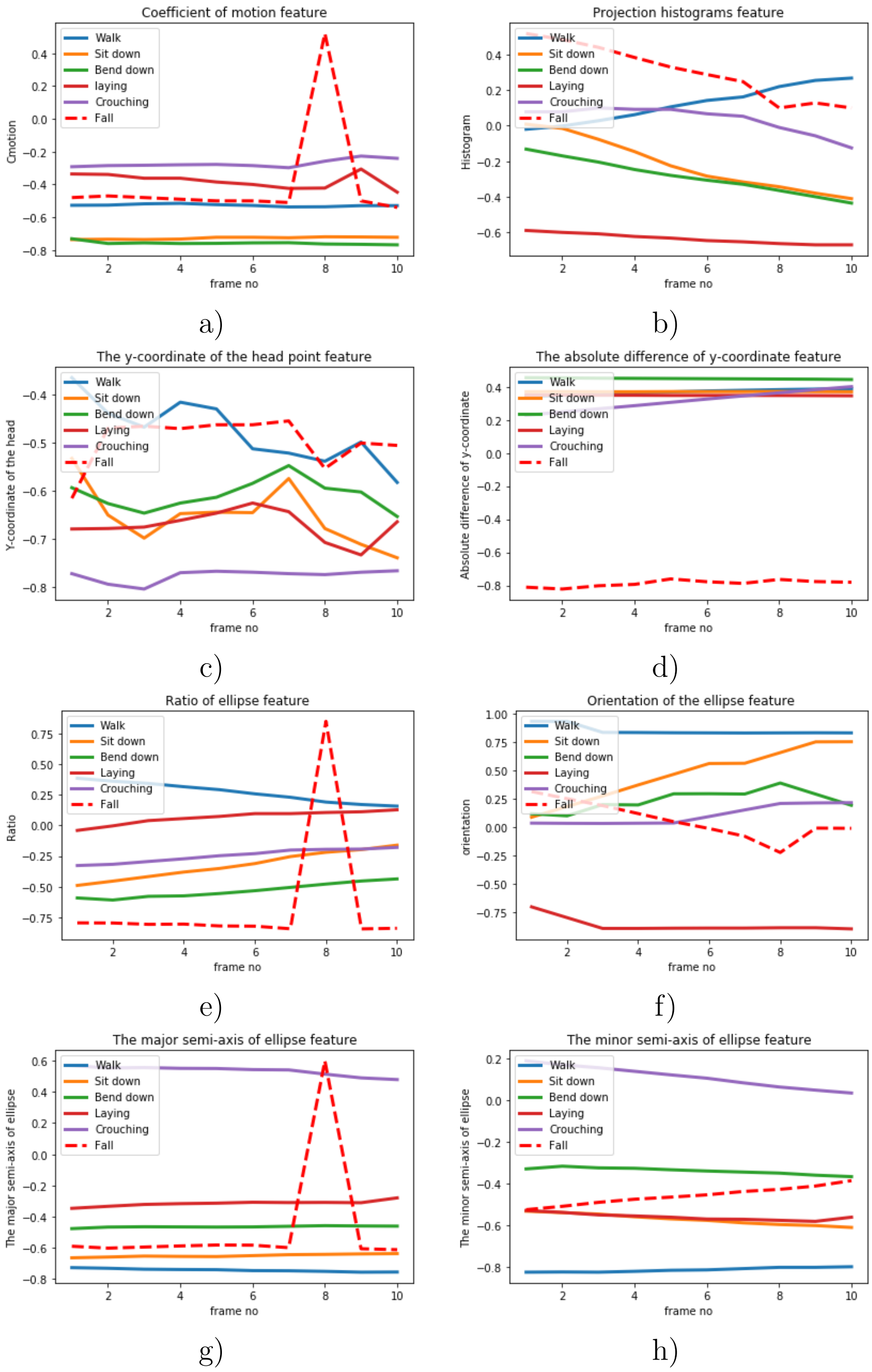

Figure 6.7: Features used to distinguish between normal daily activities and fall. Normal daily activities include walking, sitting, bending, crouching and lying down. Selected features are a) coefficient of motion b) the orientation of the ellipse c) the ratio of ellipse d) the major semi-axis e) the minor semi-axis f) the difference between the horizontal and vertical projection histograms g) the y-coordinate of the head point h) the absolute difference of y-coordinate

Chapter 7

Fall Detection Approach Using Other Machine Learning Methods

7.1 Introduction

In support of independent living for older adults, novel camera-based fall detection approaches were proposed in this research study. The proposed approach extracts 10 features from RGB images representing the motion and shape of the human body to identify falls. Motion information of a segmented silhouette can provide a useful cue for classifying different behaviours, while variation in shape and the projection histogram can be used to describe human body postures and subsequent fall events. The proposed approach represents the change of the human body using the best-fit approximated ellipse and a bounding box around the human body, produces projection histograms and determines the head position over time. These features are fed into a number of machine learning algorithms for fall classification. The proposed approach is validated with the publicly available UR Fall Detection dataset and it has outperformed many of the existing methods.

Features extracted from the human silhouette are fed into different classifiers to enhanced their performances. The classifiers used include SVM, KNN and

Bagging tree. This chapter presents the following components:

- using the selected combination of features based on the motion and the change in human shape to increase the accuracy of the proposed fall detection approach,
- training machine learning algorithms to classify falls,
- evaluating the proposed fall detection approach on publicly available dataset; UR Fall Detection dataset and the recorded video data, and compare the performance to that achieved by the state-of-the-art methods.

In Section 7.2 fall classification results are presented. The comparison with the state of the art methods is stated in Section 7.3. The conclusions of this chapter are drawn in Section 7.4

7.2 Experimental Dataset and Process

Experiments were conducted using the UR Fall Detection dataset and the recorded video data. This section illustrates the experimental details on both datasets using the proposed fall detection approach.

7.2.1 Data Pre-processing

In order to segment out the foreground object from the background, a simple background subtraction algorithm [94] was used to extract the silhouette from the background. After the silhouettes were extracted through segmentation, the second step was to extract useful features from the human silhouette to detect falls. These are the features discussed and defined in Section 4.3.

For this study, UR Fall Detection dataset is used. This dataset includes 7177 samples with 10 features. Furthermore, recorded video data from our experiments is used for further evaluation. The dataset contains 23954 with 10 features.

For both datasets, the training samples were created from the selected features vectors and stored along with the target output values corresponding to the specific classes. The fully trained machine learning classifier used on the test data to classify fall and non-fall activities.

For every video sequence, 20 frames per window with an overlap of 19 frames per window are chosen. For example, the first window contains frames from frame 1 to frame 20, the second window will contain frames from frame 2 to frame 21 and so on. The features extracted from every frame, and from every sequence of 20 frames are used to generate one feature vector which is used as an input to the classifiers for fall detection. A 10-fold cross-validation method was used in which the whole data was divided into 10 subsets, each subset is used for testing and the remainder for training. The training and testing process was repeated for all possiple combinations of the ten sets.

7.2.2 Falls Classification and Detection Techniques

This section presents the evaluation and results of the proposed fall detection approach tested with same dataset. Therefore, the experimental process were carried out using one dataset each time. The total number of features 10 are fed into one of the machine learning algorithms to classify falls.

Neural Network In order to classify between falls and non-fall activities,

Table 7.1: Recognition results of the proposed fall detection approach using UR Fall Detection dataset.

Method	Accuracy	F-Score	Presision	Sensitivity	Specitifity
NN	99.24%	99.56%	99.60%	99.52%	97.38%
SVM	93.56%	96.18%	99.45%	93.12%	96.54%
KNN	99.45%	99.68%	99.59%	99.77%	97.31%
Bagging	99.20%	99.54%	99.09%	100%	93.94%

Table 7.2: Recognition results of the proposed fall detection approach using recorded video data.

Method	Accuracy	F-Score	Presision	Sensitivity	Specitifity
NN	99.94%	99.97%	99.96%	99.97%	98.69%
SVM	99.20%	99.59%	100%	99.18%	100%
KNN	99.96%	99.98%	99.97%	99.99%	98.82%
Bagging	100%	100%	100%	100%	100%

the feature vector was fed as input to a Multilayer Perceptron Neural Network (MLPNN). The number of neurons in the hidden layer is ten. The hidden layer makes use of the scaled conjugate gradient algorithm. The outputs of the hidden layer are passed to the output layer, from where the decisive outputs are generated. The data were normalised with zero mean and standard deviation equal to 1. A mean squared error function was chosen as the evaluation criterion.

The optimal parameters were selected during the training phase (parameter tuning phase) which correspond to the maximum classification accuracy. In the case of the neural network, we varied the classifier parameters to select the optimal MLP architecture (one hidden layer with ten (10) neurons) which corresponds to the best classification rate. From Table 7.1, we can notice that the experiments with UR Fall Detection dataset shows that our approach can achieve a 99.24% accuracy, 99.52% sensitivity and 97.38% specificity. In addition, Table 7.2 indicates that our approach tested with the recorded video data provides a high accuracy of 99.94% with 99.97% and 98.69% sensitivity and specificity respectively.

Support Vector Machine The data were classified using SVM classifier. Different SVM-kernel were iteratively tested, and two hyper-parameters(sigma σ and C) and the ones that gave the highest accuracy were selected. The quadratic kernel used to separate the training and testing data. The SVM with quadratic kernels and ($\sigma = 0.125$ and $C = 128$) provides an accuracy of 93.56% tested with the UR Fall Detection dataset and 99.20% tested with recorded video data.

K Nearest Neighbour For the KNN classification, the optimal K value depends on the data. The K value is used for determining the nearest neighbours of a point. The parameter K was varied from 1 to 20 to select the optimal value. Based on the highest accuracy, the corresponding value of K is 3.

Table 7.1 shows that the KNN classifier outperforms the NN and SVM and allows us to identify the fall events with better accuracy, F-score and sensitivity on both datasets. The KNN classifier achieves the highest accuracy of 99.45%. However, NN provides better precision and specificity.

Bagging For the bagging-based method, the number of iterations (the number of generated binary decision trees) that gave the higher accuracy is 100 trees. Table 7.1 show that bagging method is superior with respect to sensitivity 100% and achieve up to 99.20% accuracy. The results presented in Table 7.2 demon-

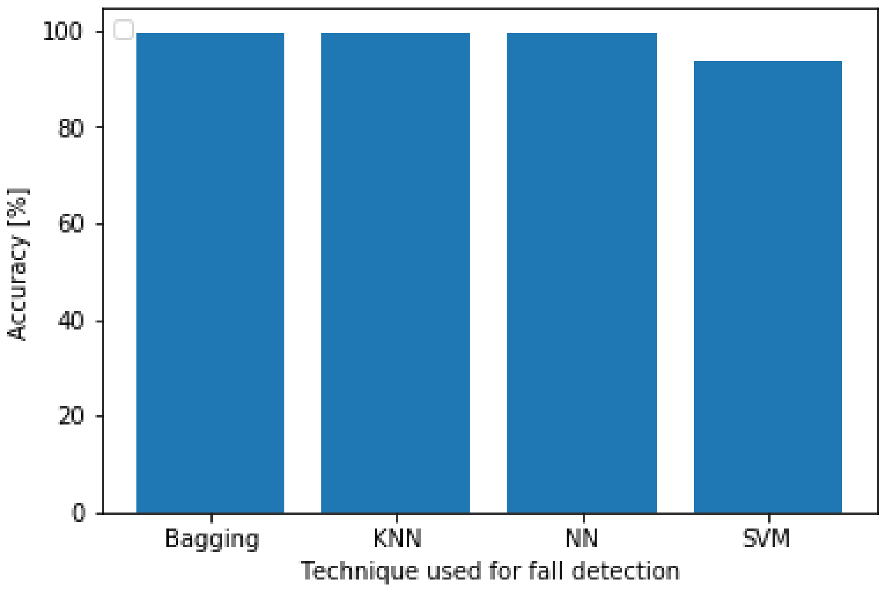

Figure 7.1: Accuracy of machine learning techniques to detect falls using UR Fall Detection.

strate that the bagging-based method outperformed all other machine leaning methods with 100%. The experimental results presented in Table 7.1 and Table 7.2 demonstrate that our selected features give promising results using NN, KNN and Bagging algorithm. In addition, the bagging tree provides the highest sensitivity which reveals its ability to detect real falls. The highest specificity value is obtained by the NN algorithm which reflects the capacity to avoid false positives (FPs), since only a reduced number of like-fall activities was confused as real falls.

Figure 7.1 shows the accuracy percentage of the proposed fall detection approach over different machine learning techniques tested with UR Fall Detection dataset. Figure 7.2 presents the accuracy percentage of the proposed fall detection approach of different machine learning techniques tested with recorded video data.

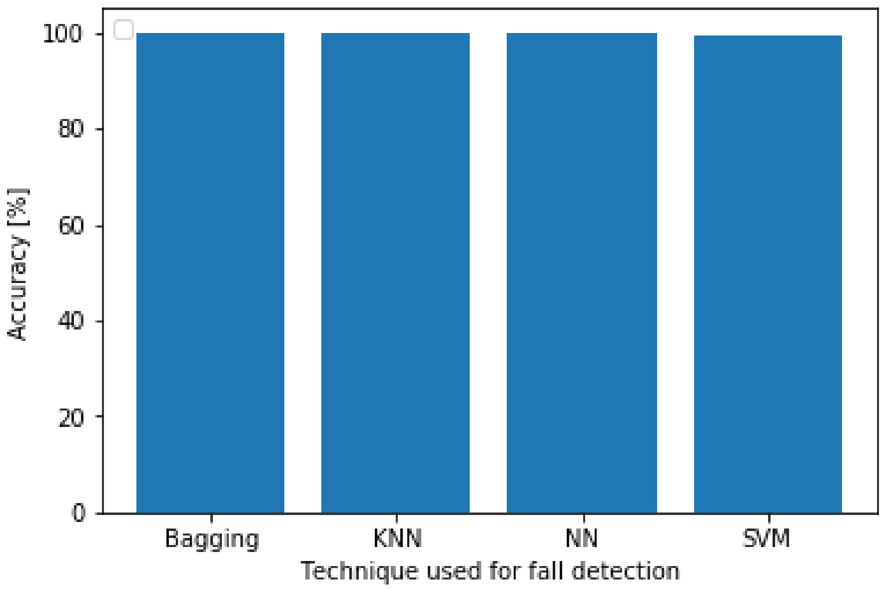

Figure 7.2: Accuracy of machine learning techniques to detect falls using recorded video data.

7.2.3 Falls Classification and Detection Techniques with a Different Dataset

While previous experiments considered training and testing the classifies on each dataset individually, this section presents experiments which do training on one dataset and testing on the other one.

In this case, all video segments from the UR Fall Detection dataset, 7177 samples with 10 features were used for training and all video segments from the recorded video data, 23954 samples with 10 features were used for testing and then vice versa. It can be noticed from Table 7.3 that the proposed fall detection approach achieves a high accuracy 99.48%, precision 99.65% and specificity 97.63% using a KNN classifier. Similarly, Table 7.4 shows that the KNN classifier outperforms other methods with accuracy up to 99.97% and precision 99.97%.

Experimental results prove that the proposed fall detection approach have a generatic feature. Even though, the approach has been tested in two different datasets, obtaining state of the art results on both of them. To the best of the knowledge, the proposed approach is the first one achieving such results on different datasets. Notice that both datasets present different characteristics as explained in Section 4.4.1. The UR Fall Detection dataset contains totally different scenarios in contrast to the recorded video data. Although, there are significant differences among both dataets, the proposed approach achieves good results, which could be considered as a solid proof of the generality of the proposed fall detection approach. The KNN classifier outperforms other machine learning algorithms in terms of accuracy and precision.

7.3 Comparison with the State of the Art Methods

The proposed fall detection approach was compared with the most cited fall detection approaches in the literature. Related results on UR Fall Detection dataset using only RGB images are shown in Table 7.5.

Min et al [77] propose a fall detection method for falls against furniture using a R-CNN to obtain the information of locations and objects in the scene. Experiments on UR Fall detection dataset achieves an accuracy of 95.50%,where, the AUROC was 94%. However, the proposed approach is accurate in predicting

Table 7.3: Recognition results of the proposed fall detection approach using UR Fall Detection dataset for the training and recorded video data for testing

Method	Accuracy	F-Score	Precision	Sensitivity	Specificity
NN	98.89%	99.36%	99.31%	99.41%	95.32%
SVM	96.22%	98.03%	97.11%	98.96%	44.80%
KNN	99.48%	99.70%	99.65%	99.76%	97.63%
Bagging	98.89%	99.36%	98.89%	99.84%	92.47%

Table 7.4: Recognition results of the proposed fall detection approach using Recorded video data for the training and UR Fall Detection dataset for testing.

Method	Accuracy	F-Score	Precision	Sensitivity	Specificity
NN	99.90%	99.95%	99.94%	99.96%	97.77%
SVM	96.24%	98.05%	96.59%	99.55%	34.14%
KNN	99.97%	99.98%	99.97%	100%	88.97%
Bagging	99.83%	99.91%	99.83%	100%	93.85%

non-fall activities, with a detection rate of 99.45%.

Marcos et al. [81] employ a CNN to detect falls. Their method obtains an accuracy of 95%, sensitivity and specificity were 100% and 92% respectively. Adopting a CNN, yields better sensitivity than the proposed approach. On the other hand, its specificity is low which means that some daily activities are misclassified as falls.

The proposed fall detection system performs better than the compared deep learning approaches presented in [77] and [81]. The reason may be that deep learning approaches require large training datasets to achieve good performances. Due to the limited training data obtained from UR Fall Detection dataset, the proposed soluation was more suitable than the deep learning approaches. In addition, deep learning approaches are computationally expensive as they require expensive GPUs and fast CPU to train large datasets, containing millions of images. Moreover, machine learning algorithms are easier to tune as deep learning is usually used as a black box.

Kwolek and Kepski [62] present an embedded system for fall detection based on acceleration data and depth maps. The system achieves an accuracy of 90%, precision of 83.30%, sensitivity 100% and specificity 80%.

It is worth mentioned that the above fall detection approaches combine information from several cameras or embedded systems which involve information from both video data and acceleration data extracted from an accelerometer. While, in the proposed approach only a single camera is used.

Table 7.5: Comparison of Existing Methods using the same database. Where no values were given by the authors, the fields are blank.

Method	Accuracy	Precision	Sensitivity	Specificity
R-CNN [77]	95.50%	–%	–	–
Convolutional NN [81]	95%	–	100%	92%
SVM [62]	90%	83.30%	100%	80%
SVM [50]	96.66%	93.55	100%	94.93%
SVM [122]	99.83%	–	100%	100%
SVM/Adaboost [25]	99.42%	95.91%	92.15%	99.79%
SVM [110]	–	–	93.70%	92.00%
Proposed fall detection approach	99.45%	99.59%	99.77%	97.31%

Harrou et al. [50] introduce a fall detection approach based on the combination of a (MEWMA) and a SVM classifier. The results report an accuracy of 96.66%, precision of 93.55%, sensitivity of 100% and specificity of 94.93%.

Yun and Gu. [122] propose a method for fall detection through analysing dynamic shape and motion of human body regions on Riemannian manifolds. The results show a high detection rate of 99.83% with sensitivity of 100% and specificity of 100%.

Charfi et al. [25] present a fall detection method using SVM and Adaptive Boosting algorithms. The results show an accuracy of 99.42%, precision of 95.91%, sensitivity of 92.15% and specificity of 99.79%.

Wang et al. [110] propose fall detection method using multiple cameras and a SVM classifier to distinguish fall and non-fall activities. Their method achieves 93.70% sensitivity and 92.0% specificity. Their experiments are performed using 30 frames per window, where the proposed approach outperforms their results by using only 20 frames per window. More precisely, the proposed system obtained a sensitivity of 99.77% and a specificity of 97.31%, while Wang et al achieves only 93.70% and 92.0% for the same metrics, respectively.

The proposed approach exhibited an accuracy higher than all other approaches except the method presented in [122]. The results presented in Table 7.5 show that the proposed fall detection approach highly improves the classification performance. The proposed fall detection approach is about 9% more accurate than [62] and 5% more accurate than [77] and [81].

The experimental results demonstrate that the proposed fall detection approach obtained one of the best fall detection methods in the literature. Even though using of a simple extracted feature, the results are outstanding as compared to the fall detection approaches that use similar or advanced machine learning algorithms. It is considered that the main reason behind the success of the proposed approach is the selected features are robust and effective to discriminate between fall and non-fall activities.

The results in Table 7.5, show that the proposed fall detection approach in this thesis performs better than all other methods by a large margin on the aspects of the precision. This demonstrates that the proposed fall detection approach can effectively predict falls. The proposed method achieves a precision of 99.59%, while the highest value of the precision of other methods is 95.91%. Moreover, the proposed approach achieves the best specificity of 97.31% among all methods except two methods [122] and [25]. This means that the proposed approach can classify most daily activities correctly as non-fall activities.

7.4 Discussion

The work presented in this study has mainly focused on evaluating the proposed fall detection approach based on computer vision techniques. The approach presented here employs enhanced features which are extracted from human silhouettes. Then the features are fed into machine learning algorithms to classify falls. The experimental results show that the proposed algorithm using these features is reliable for fall detection. The results also demonstrate that the proposed set of features achieves good performance tested with various machine learning classifiers on a publicly available dataset. The KNN achieves better classification performance in terms of accuracy 99.45% comparing to the other methods. The ANN provides a better precision 99.60% and specificity 97.38% and this reflects its capability to correctly classifying daily activities and fall events.

Chapter 8

Conclusions and Future Works

8.1 Thesis Summary

The work presented in this thesis is a novel attempt to automatically detect fall incidents to support independent living for older adults in indoor environments. Based on the results obtained from the used techniques, it can be concluded that the ability to distinguish a fall action depends mainly on the quality of the classifier inputs. Therefore, the features of the extracted human silhouette play a key role in the effectiveness and robustness of detecting human falls. The proposed approaches were based on a combination of visual features to decide if a fall has happened in a video sequence.

The aim of this research was to investigate appropriate methods for detecting falls through analysing the motion and the shape of the human body. To achieve the project aim, a video-based system for detecting falls is proposed. The proposed system consists of three steps: detection of moving objects in a frame, tracking such objects from frame to frame and then analysis of object tracks to understand the behaviour.

The video data provided for testing the proposed fall detection approaches were collected from real environments as well as from publicly available datasets. The recorded video data contains simulated falls and daily activities collected in realistic situations. The reason for collecting video data was to provide enough experimental data for this research. The data was obtained from the visual recording of an indoor environment where participants simulated falls and performed usual activities in a domestic setting. The data was acquired using a static camera. The data is needed to enable the development of methods to automatically identify and distinguish between activity associated with daily living of occupants and that associated with abnormal behaviour such as fall events.

The video data sequences are used to extract robust features which describe the change in human shape and to discriminate falls from other activities like lying and sitting. These features are based on motion, changes in the human shape, projection histogram and temporal change of head position. The features extracted from the human silhouette are finally fed into various machine learning classifiers for fall detection. The experimental results proved that the proposed approach clearly discriminate a fall event from other daily activities.

In summary, throughout this research, original knowledge on visual-based human fall detection was gained. The research conclusions with critical discussion and the direction for future work are presented in the remaining sections of this chapter.

8.2 Concluding Remarks

This thesis attempts to provide fall detection approaches for supporting older adults in indoor environments. The conclusions for the various aspects of the project are presented below:

8.2.1 Fall Detection Approach Using Threshold-Based Methods

This thesis presents novel threshold-based approaches to detect falls for an older person in a home environment. The first approach combines a motion feature and a change of the orientation and the ratio features to detect a fall event. The second approach employs three features; motion information, human shape variation and projection histogram to detect a fall.

The first approach investigated the effectiveness of motion information extracted by the tMHI method, using variation in shape and fits an approximated ellipse around the human body to further improve the accuracy of fall detection. The second approach proposes a novel feature based on computing the standard deviation of the difference of the horizontal and the vertical histograms. The proposed fall detection approach based on a combination of timed motion history, variation in human shape and projection histograms achieves an accuracy of 99.82% while maintaining small false alarms.

Threshold-based fall detection methods are simple and cost-effective methods. However, the performance heavily depends on the fixed threshold level. For each dataset, it is necessary to determine threshold values that are the best to detect falls. Moreover, threshold values need to be adapted for monitoring different people.

8.2.2 Fall Detection Approach Using Neural Network

This thesis highlights the need for extracting a number of features from binary human silhouettes for fall detection. For each video sequence, the moving object is detected, and 10 unique features represented in the feature vector F are used to identify falls. These features are coefficient of motion, the orientation of the ellipse, the ratio of the ellipse, the major semi-axis, the minor semi-axis of the ellipse, the difference between the horizontal and vertical projection histograms, the y-coordinate of the head point, and the standard deviation of the absolute difference of y-coordinate. These features are fed into a multilayer perceptron neural network for fall classification.

Experimental results show efficiency and reliability of the proposed fall detection approach with high fall detection rate of 99.60% tested with UR Fall Detection dataset. Additionally, A set of experiments have been conducted using our recording dataset, the results indicate that the proposed approach achieves high fall detection rate 99.94% and low false alarm 0.02%.

The overall performance of MLP Neural Network was better than the performance of threshold-based algorithms. The MLP Neural Network achieves an accuracy of 99.94%, while threshold-based methods achieves 99.82% tested with recording dataset.

8.2.3 Fall Detection Approach Using Various Machine Learning Methods

This thesis also shows that the extracted features allow discrimination of falls from other activities like lying and sitting. These features are based on motion, change in the human shape, projection histogram and temporal change of head position. The proposed fall detection approach is evaluated by other machine learning techniques such as, support vector machine, K-Nearest Neighbour and Bagging tree. Experimental results show that the proposed algorithm using these features is reliable for fall detection. We demonstrate that the proposed set of features achieves good performance with various machine learning classifiers on publicly available dataset. Additionally, the results and knowledge gained from this research demonstrate that the NN provides a better precision 99.60% and specificity 97.38% and this reflects its capability to correctly classify activities like falls as non-fall events. The proposed system is able to successfuly classify the daily activities from falls.

In general, the major findings of this work in terms of detecting falls from visual data, and using various computer vision techniques are listed below.

1. Timed Motion History Images provide coherent motion information to represent the motion trail of a moving object over time.

2. The use of the major semi-axis and minor semi-axis of the ellipse fitted around the human silhouette as features increases the accuracy of the proposed fall detection.

3. Introducing a new feature based on computing the maximum values of foreground pixels in the horizontal and vertical histograms, as well as the difference between maximum values helps to effectively discriminate fall among other activities.

4. A new feature extracted from the person's head was explored. The difference of the y-coordinate of the person head as well as the standard deviation of the difference of y-coordinate of the person head improves the accuracy and the sensitivity of the proposed fall detection approach.

5. Identifying the best combinations of features capable of effectively detecting a fall.

6. The demonstrated results indicate that machine learning methods outperformed the threshold methods for fall detection. Additionally, the results presented in this research show that the selected features perform well with various machine learning techniques.

7. The results demonstrate that using the right selection of features can outperform more heavily computational techniques (e.g Convolution neural network).

In the remaining part of this chapter, some future work is proposed.

8.3 Future Work

In a real-life situation many different factors affect the performance of fall detection system. First of all, in real environments background is changed when added static object or move a piece of furniture. Therefore, more advanced foreground detection techniques using Convolution Neural Network [77] could be implemented for more reliable system performance.

The proposed fall detection system is implemented for monitoring a single person at home and is not adequate for monitoring multiple people. According to [118] people counting techniques can be used to counting number of people. Firstly, moving people are detected as foreground regions and then each person is tracked through consecutive frames using a correlation-based algorithm. The other scenario when an older adult has a pet, in this case only the extracted foreground region corresponding to the human body silhouette are used for fall detection. The extracted foreground region can be determined by using object classification techniques.

The proposed fall detection system can only work in the presence of a good intensity of light and further improvements are needed to deal with poor lighting conditions such as lighting changes, different type of light source, and dark environment. A possible salutation is to use low-cost infrared systems detecting moving object. In addition, occlusion in the indoor environment is another problem which can be addressed by using multiple cameras to make sure that the human body is visible in at least one camera view.

There are several lines of research arising from this work which should be pursued. Some suggetions are listed below.

- Further improvements on the segmentation algorithm could be one of future tasks of this research. A more efficient background subtraction algorithm could be applied for a better human silhouette segmentation.

- In order to increase the robustness of the proposed fall detection approach, multiple cameras in a single scene could be another suggestion for the future work. Although the achieved results for fall detection are encouraging using only a single camera, the classification performance could be improved further by using multiple cameras, since a single camera limits the view angle of the scene.

- From information aspects, instead of using only the video data, some data from additional sensors like accelerometer could be extracted, with the aim of increasing the detection rate. Several sensors have been shown to give a better result when combined with camera sensor. These types of sensors will be considered for future work. However, this work is already a step forward towards high-performance fall detection systems.

- Using the 3D information from a depth camera like the Microsoft Kinect to obtain additional depth information for a better human silhouette segmentation could be studied.

- Defining normal inactivity zones such as bed, chair or other typical furniture in the scene could decrease the false positives and give more robustness for the fall detection system.

- It will be interesting to extend the work to develop deep learning approaches. The deep neural network extracts the most discriminate features from consequent images based on the training data, and hence, covers more real-life scenarios.

Fall recognition is a positive step in support of independent living. However, the goal of older adults safety and fall prevention to be achieved.

References

[1] Boosting and bagging: How to develop a robust machine learning algorithm. available at:. https://hackernoon.com/how-to-develop-a-robust-algorithm. Accessed: 2018-10-02. xvi, 68

[2] Knn classification using scikit-learn learn k-nearest neighbor(knn) classification and build knn classifier using python scikit-learn package. available at:. https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn. Accessed: 2018-10-02. xvi, 67

[3] National population projections - office for national statistics. available at:. https://www.ons.gov.uk. Accessed: 2018-08-24. 1

[4] support vector machine-introduction to machine learning. available at:. https://towardsdatascience.com. Accessed: 2018-10-02. xvi, 66

[5] World health organisation. who global report on fall prevention in older age . available at:. http://www.who.int/news-room/fact-sheets/detail/falls. Accessed: 2018-06-24. 9

[6] Stefano Abbate, Marco Avvenuti, Francesco Bonatesta, Guglielmo Cola, Paolo Corsini, and Alessio Vecchio. A smartphone-based fall detection system. *Pervasive and Mobile Computing*, **8**[6]:883–899, 2012. 10

[7] Md Atiqur Rahman Ahad. *Motion history images for action recognition and understanding*. Springer Science & Business Media, 2012. 53

[8] Md Atiqur Rahman Ahad, Joo Kooi Tan, Hyoungseop Kim, and Seiji Ishikawa. Motion history image: its variants and applications. *Machine Vision and Applications*, **23**[2]:255–281, 2012. 53, 54, 55

[9] Suad Albawendi, Kofi Appiah, Heather Powell, and Ahmad Lotfi. Video based fall detection with enhanced motion history images. In *Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '16, pages 29:1–29:7, 2016. 52, 80, 92

[10] Suad Albawendi, Kofi Appiah, Heather Powell, and Ahmad Lotfi. Video based fall detection using features of motion, shape and histogram. In *Proceedings of the 11th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '18, pages 529–536, New York, NY, USA, 2018. ACM. 92, 97

[11] Laila Alhimale, Hussein Zedan, and Ali Al-Bayatti. The implementation of an intelligent and video-based fall detection system using a neural network. *Applied Soft Computing*, **18**:59–69, 2014. 63

[12] Syed Ali, Reamsha Khan, Arif Mahmood, Malik Hassan, and Moongu Jeon. Using temporal covariance of motion and geometric features via boosting for human fall detection. *Sensors*, **18**[6]:1918, 2018. 18

[13] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2014. 66

[14] Maryam Asrar, Amin Al-Habaibeh, and Mohammed Houda. Innovative algorithm to evaluate the capabilities of visual, near infrared, and infrared technologies for the detection of veins for intravenous cannulation. *Applied Optics*, **55**[34]:D67–D75, 2016. 14

[15] Edouard Auvinet, Caroline Rougier, Jean Meunier, Alain St-Arnaud, and Jacqueline Rousseau. Multiple cameras fall dataset. *DIRO-Universit de Montral, Tech.Rep*, **1350**, 2010. 18

[16] Ke-Jia Bai. A new object tracking algorithm based on mean shift in 4-d state space and on-line feature selection. In *Information and Computing*

(ICIC), Third International Conference on, **1**, pages 39–42. IEEE, 2010. 50

[17] Angela Barriga, Jos M. Conejero, Juan Hernndez, Elena Jurado, Enrique Moguel, and Fernando Snchez-Figueroa. A vision-based approach for building telecare and telerehabilitation services. *Sensors*, **16**[10]:1724, 2016. 13

[18] Zhen-Peng Bian, Junhui Hou, Lap-Pui Chau, and Nadia Magnenat-Thalmann. Fall detection based on body part tracking using a depth camera. *IEEE journal of biomedical and health informatics*, **19**[2]:430–439, 2015. 13, 21

[19] F. Bloch, V. Gautier, N. Noury, J-E Lundy, J. Poujaud, Y-E Claessens, and A-S Rigaud. Evaluation under real-life conditions of a stand-alone fall detector for the elderly subjects. *Annals of physical and rehabilitation medicine*, **54**[6]:391–398, 2011. 37

[20] Gary R. Bradski and James W. Davis. Motion segmentation and pose recognition with motion history gradients. *Machine Vision and Applications*, **13**[3]:174–184, 2002. 53

[21] Leo Breiman. Bagging predictors. *Machine Learning*, **24**[2]:123–140, 1996. 67

[22] Kabalan Chaccour, Rony Darazi, Amir Hajjam El Hassani, and Emmanuel Andrs. From fall detection to fall prevention: A generic classification of fall-related systems. *IEEE Sensors Journal*, **17**[3]:812–822, 2017. 2, 59

[23] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, **2**[3]:27, 2011. 65

[24] Imen Charfi, Johel Miteran, Julien Dubois, Mohamed Atri, and Rached Tourki. Definition and performance evaluation of a robust svm based fall detection solution. *SITIS*, **12**:218–224, 2012. 20

[25] Imen Charfi, Johel Miteran, Julien Dubois, Mohamed Atri, and Rached Tourki. Optimized spatio-temporal descriptors for real-time fall detection: comparison of support vector machine and adaboost-based classification. *Journal of Electronic Imaging*, **22**[4]:041106, 2013. xv, 7, 28, 30, 40, 43, 56, 114, 115, 116

[26] Qiang Chen, Quan-Sen Sun, Pheng Ann Heng, and De-Shen Xia. Two-stage object tracking method based on kernel and active contour. *IEEE Transactions on Circuits and Systems for Video Technology*, **20**[4]:605–609, 2010. 50

[27] Saisakul Chernbumroong, Ahmad Lotfi, and Caroline Langensiepen. Prediction of mobility entropy in an ambient intelligent environment. In *Intelligent Agents (IA), IEEE Symposium on*, pages 65–72. IEEE, 2014. 12

[28] Claudio Coppola, Serhan Cosar, Diego R. Faria, and Nicola Bellotto. Automatic detection of human interactions from rgb-d data for social activity classification. In *Robot and Human Interactive Communication (RO-MAN), 26th IEEE International Symposium on*, pages 871–876. IEEE, 2017. 19, 39

[29] Nello Cristianini and John Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000. 65

[30] Xiaoxiao Dai, Meng Wu, Bradley Davidson, Mohammad Mahoor, and Jun Zhang. Image-based fall detection with human posture sequence modeling. In *Healthcare Informatics (ICHI), 2013 IEEE International Conference on*, pages 376–381. IEEE, 2013. 19

[31] Deepjoy Das and Dr Saharia. Implementation and performance evaluation of background subtraction algorithms. *arXiv preprint arXiv:1405.1815*, 2014. 27, 45

[32] Koldo de Miguel, Alberto Brunete, Miguel Hernando, and Ernesto Gambao. Home camera-based fall detection system for the elderly. *Sensors*, **17**[12]:2864, 2017. 11, 12, 23, 52, 92

[33] Glen Debard, Peter Karsmakers, Mieke Deschodt, Ellen Vlaeyen, Eddy Dejaeger, Koen Milisen, Toon Goedem, Bart Vanrumste, and Tinne Tuytelaars. *Camera-based fall detection on real world data*, pages 356–375. Outdoor and large-scale real-world scene analysis. Springer, 2012. 55

[34] Glen Debard, Peter Karsmakers, Mieke Deschodt, Ellen Vlaeyen, Jonas Van den Bergh, Eddy Dejaeger, Koen Milisen, Toon Goedem, Tinne Tuytelaars, and Bart Vanrumste. Camera based fall detection using multiple features validated with real life video. In *Workshop Proceedings of the 7th International Conference on Intelligent Environments*, **10**, pages 441–450. IOS Press, 2011. 20

[35] Barga Deori and Dalton Meitei Thounaojam. A survey on moving object tracking in video. *International Journal on Information Theory (IJIT)*, **3**[3]:31–46, 2014. 47, 48

[36] Wenwen Ding, Kai Liu, Hao Chen, and Fengqin Tang. Human action recognition using similarity degree between postures and spectral learning. *IET Computer Vision*, **12**[1]:110–117, 2017. 14

[37] Rui dong Wang, Yong liang Zhang, Ling ping Dong, Jia wei Lu, Zhi qin Zhang, and Xia He. Fall detection algorithm for the elderly based on human characteristic matrix and svm. In *Control, Automation and Systems (ICCAS), 15th International Conference on*, pages 1190–1195. IEEE, 2015. 12

[38] Anastasios Doulamis and Nikolaos Doulamis. Adaptive deep learning for a vision-based fall detection. In *Proceedings of the 11th PErvasive Technologies Related to Assistive Environments Conference*, pages 558–565. ACM, 2018. 25

[39] Nikolaos Doulamis. Iterative motion estimation constrained by time and shape for detecting persons' falls. In *Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments*, page 62. ACM, 2010. 25

[40] Milad Elgargni, Amin Al-Habaibeh, and Ahmad Lotfi. Cutting tool tracking and recognition based on infrared and visual imaging systems using principal component analysis (pca) and discrete wavelet transform (dwt) combined with neural networks. *The International Journal of Advanced Manufacturing Technology*, **77**[9-12]:1965–1978, 2015. 15

[41] Weiguo Feng, Rui Liu, and Ming Zhu. Fall detection for elderly person care in a vision-based home surveillance environment using a monocular camera. *Signal, Image and Video Processing*, **8**[6]:1129–1138, 2014. 27, 52, 56, 59, 74, 79, 91

[42] J. Fleming, C. Brayne, and Cambridge City over-75s Cohort (CC75C) study collaboration. Inability to get up after falling, subsequent time on floor, and summoning help: prospective cohort study in people over 90. *BMJ (Clinical research ed.)*, **337**:a2227, Nov 17 2008. 1, 9

[43] Homa Foroughi, Baharak Shakeri Aski, and Hamidreza Pourreza. Intelligent video surveillance for monitoring fall detection of elderly in home environments. In *Computer and Information Technology. ICCIT . 11th International Conference on*, pages 219–224. IEEE, 2008. 11, 60, 76

[44] Homa Foroughi, Alireza Rezvanian, and Amirhossien Paziraee. Robust fall detection using human shape and multi-class support vector machine. In *Computer Vision, Graphics & Image Processing. ICVGIP'08. Sixth Indian Conference on*, pages 413–420. IEEE, 2008. 10, 56

[45] Alberto Brunete Gonzlez, Micheline Selmes, and Jacques Selmes. Can smart homes extend people with alzheimers disease stay at home? *Journal of Enabling Technologies*, **11**[1]:6–12, 2017. 1

[46] Kishnaprasad G. Gunale and Prachi Mukherji. Fall detection using k-nearest neighbor classification for patient monitoring. In *Information Processing (ICIP), International Conference on*, pages 520–524. IEEE, 2015. 56

[47] M. Amac Guvensan, A. Oguz Kansiz, N. Cihan Camgoz, H. Turkmen, A. Gokhan Yavuz, and M. Elif Karsligil. An energy-

efficient multi-tier architecture for fall detection on smartphones. *Sensors*, **17**[7]:1487, 2017. 11

[48] Mabrouka Hagui, Mohamed Ali Mahjoub, and Faycel Elayeb. Coupled hidden markov model for video fall detection. In *Natural Computation (ICNC), 11th International Conference on*, pages 675–679. IEEE, 2015. 13, 54, 84

[49] Fouzi Harrou, Nabil Zerrouki, Ying Sun, and Amrane Houacine. Statistical control chart and neural network classification for improving human fall detection. In *Modelling, Identification and Control (ICMIC), 8th International Conference on*, pages 1060–1064. IEEE, 2016. 24

[50] Fouzi Harrou, Nabil Zerrouki, Ying Sun, and Amrane Houacine. Vision-based fall detection system for improving safety of elderly people. *IEEE Instrumentation Measurement Magazine*, **20**[6]:49–55, 2017. 20, 65, 114, 115

[51] Youssef Hbali, Sara Hbali, Lahoucine Ballihi, and Mohammed Sadgal. Skeleton-based human activity recognition for elderly monitoring systems. *IET Computer Vision*, **12**[1]:16–26, 2017. 13

[52] Farhad Hossain, Md Liakot Ali, Md Zahurul Islam, and Hossen Mustafa. A direction-sensitive fall detection system using single 3d accelerometer and learning classifier. In *Medical Engineering, Health Informatics and Technology (MediTec), 2016 International Conference on*, pages 1–6. IEEE, 2016. 16

[53] Yi-Zeng Hsieh and Yu-Lin Jeng. Development of home intelligent fall detection iot system based on feedback optical flow convolutional neural network. *IEEE Access*, **6**:6048–6057, 2018. 25

[54] Raul Igual, Carlos Medrano, and Inmaculada Plaza. Challenges, issues and trends in fall detection systems. *Biomedical engineering online*, **12**[1]:66, 2013. 15

[55] Ms Jyoti J. Jadhav and J. Jyoti. Moving object detection and tracking for video survelliance. *International Journal of Engineering Research and General Science*, **2**[4]:372–378, 2014. 45, 46

[56] Dean M. Karantonis, Michael R. Narayanan, Merryn Mathie, Nigel H. Lovell, and Branko G. Celler. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *Information Technology in Biomedicine, IEEE Transactions on*, **10**[1]:156–167, 2006. 10

[57] Michal Kepski and Bogdan Kwolek. Fall detection using ceiling-mounted 3d depth camera. In *2014 International conference on computer vision theory and applications (VISAPP)*, pages 640–647. IEEE, 2014. 10, 66

[58] Michal Kepski and Bogdan Kwolek. Event-driven system for fall detection using body-worn accelerometer and depth sensor. *IET Computer Vision*, 2017. 22

[59] Satoshi Kido, Tomoya Miyasaka, Toshiaki Tanaka, Takao Shimizu, and Tadafumi Saga. Fall detection in toilet rooms using thermal imaging sensors. In *System Integration, 2009. SII 2009. IEEE/SICE International Symposium on*, pages 83–88. IEEE, 2009. 14

[60] Xiangbo Kong, Lin Meng, and Hiroyuki Tomiyama. Fall detection for elderly persons using a depth camera. In *Advanced Mechatronic Systems (ICAMechS), International Conference on*, pages 269–273. IEEE, 2017. 16

[61] Panagiotis Kostopoulos, Tiago Nunes, Kevin Salvi, Michel Deriaz, and Julien Torrent. F2d: A fall detection system tested with real data from daily life of elderly people. In *17th International Conference on E-health Networking, Application & Services (HealthCom)*, pages 397–403. IEEE, 2015. 5, 10, 52, 79, 92

[62] Bogdan Kwolek and Michal Kepski. Human fall detection on embedded platform using depth maps and wireless accelerometer. *Computer methods and programs in biomedicine*, **117**[3]:489–501, 2014. xv, 7, 19, 27, 39, 41, 95, 114, 115

[63] Bogdan Kwolek and Michal Kepski. Improving fall detection by the use of depth sensor and accelerometer. *Neurocomputing*, **168**:637–645, 2015. 22, 66

[64] Jung Keun Lee, Stephen N. Robinovitch, and Edward J. Park. Inertial sensing-based pre-impact detection of falls involving near-fall scenarios. *IEEE transactions on neural systems and rehabilitation engineering*, **23**[2]:258–266, 2015. 15, 71

[65] Wanqing Li, Zhengyou Zhang, and Zicheng Liu. Action recognition based on a bag of 3d points. In *Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE Computer Society Conference on*, pages 9–14. IEEE, 2010. 14

[66] Zhiyong Li, Pengfei Li, Xiaoping Yu, and Mervat Hashem. Real-time tracking by double templates matching based on timed motion history image with hsv feature. *The Scientific World Journal*, **2014**, 2014. pmid:24592185. 27, 51, 53, 54, 73, 79, 91

[67] Rui Liang, Lei Yan, Pengqi Gao, Xu Qian, Zhongjian Zhang, and Huabo Sun. Aviation video moving-target detection with inter-frame difference. In *Image and Signal Processing (CISP), 2010 3rd International Congress on*, **3**, pages 1494–1497. IEEE, 2010. 46

[68] Chia-Wen Lin and Zhi-Hong Ling. Automatic fall incident detection in compressed video for intelligent homecare. In *Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on*, pages 1172–1177. IEEE, 2007. 1

[69] Chih-Yang Lin, Shang-Ming Wang, Jia-Wei Hong, Li-Wei Kang, and Chung-Lin Huang. Vision-based fall detection through shape features. In *Multimedia Big Data (BigMM), IEEE Second International Conference on*, pages 237–240. IEEE, 2016. 56

[70] Chien-Liang Liu, Chia-Hoang Lee, and Ping-Min Lin. A fall detection system using k-nearest neighbor classifier. *Expert Systems with Applications*, **37**[10]:7174–7181, 2010. 10, 59

[71] Mitja Lutrek and Botjan Kalua. Fall detection and activity recognition with machine learning. 2008. 68

[72] Kristna Machov, Frantisek Barcak, and Peter Bednr. A bagging method using decision trees in the role of base classifiers. *Acta Polytechnica Hungarica*, **3**[2]:121–132, 2006. 68

[73] Sawsan M. Mahmoud, Ahmad Lotfi, and Caroline Langensiepen. Abnormal behaviours identification for an elder's life activities using dissimilarity measurements. In *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments*, page 25. ACM, 2011. 10, 69

[74] Aihua Mao, Xuedong Ma, Yinan He, and Jie Luo. Highly portable, sensor-based system for human fall monitoring. *Sensors*, **17**[9]:2096, 2017. 10

[75] Georgios Mastorakis and Dimitrios Makris. Fall detection system using kinects infrared sensor. *Journal of Real-Time Image Processing*, **9**[4]:635–646, 2014. 16

[76] S-G Miaou, Pei-Hsu Sung, and Chia-Yuan Huang. A customized human fall detection system using omni-camera images and personal information. In *Distributed Diagnosis and Home Healthcare. D2H2. 1st Transdisciplinary Conference on*, pages 39–42. IEEE, 2006. 13

[77] Weidong Min, Hao Cui, Hong Rao, Zhixun Li, and Leiyue Yao. Detection of human falls on furniture using scene analysis based on deep learning and activity characteristics. *IEEE Access*, **6**:9324–9335, 2018. 25, 113, 114, 115, 121

[78] Behzad Mirmahboub, Shadrokh Samavi, Nader Karimi, and Shahram Shirani. Automatic monocular system for human fall detection based on variations in silhouette area. *Biomedical Engineering, IEEE Transactions on*, **60**[2]:427–436, 2013. 10, 12

[79] Martin Fodslette Mller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, **6**[4]:525–533, 1993. 65

[80] Urbano Miguel Nunes, Diego R. Faria, and Paulo Peixoto. A human activity recognition framework using max-min features and key poses with differential evolution random forests classifier. *Pattern Recognition Letters*, **99**:21–31, 2017. 17

[81] Adrin Nez-Marcos, Gorka Azkune, and Ignacio Arganda-Carreras. Vision-based fall detection with convolutional neural networks. *Wireless Communications and Mobile Computing*, **2017**, 2017. 24, 114, 115

[82] Olukunle Ojetola. *Detection of human falls using wearable sensors*, 2013. 39

[83] Olukunle Ojetola, Elena Gaura, and James Brusey. Data set for fall events and daily activities from inertial sensors. In *Proceedings of the 6th ACM multimedia systems conference*, pages 243–248. ACM, 2015. 23

[84] CI Patel and Ripal Patel. Gaussian mixture model based moving object detection from video sequence. In *Proceedings of the International Conference Workshop on Emerging Trends in Technology*, pages 698–702. ACM, 2011. 48

[85] Sandeep Kumar Patel and Agya Mishra. Moving object tracking techniques: A critical review. *Indian Journal of Computer Science and Engineering*, **4**[2]:95–102, 2013. 49

[86] IPES Putra, James Brusey, Elena Gaura, and Rein Vesilo. An event-triggered machine learning approach for accelerometer-based fall detection. *Sensors*, **18**[1]:20, 2017. 16, 23

[87] Hamid Rajabi and Manoochehr Nahvi. An intelligent video surveillance system for fall and anesthesia detection for elderly and patients. In *Pattern Recognition and Image Analysis (IPRIA), 2nd International Conference on*, pages 1–6. IEEE, 2015. 13, 55, 71, 88, 103

[88] S. Ravi and AM Khan. Morphological operations for image processing: understanding and its applications. In *Proc. 2nd National Conference on VLSI, Signal processing Communications NCVSComs-2013*, 2013. 49

[89] Henry Rimminen, Juha Lindstrm, Matti Linnavuo, and Raimo Sepponen. Detection of falls among the elderly by a floor sensor using the electric near field. *IEEE Transactions on Information Technology in Biomedicine*, **14**[6]:1475–1476, 2010. 13

[90] Caroline Rougier, Edouard Auvinet, Jacqueline Rousseau, Max Mignotte, and Jean Meunier. Fall detection from depth map video sequences. In *International Conference on Smart Homes and Health Telematics*, pages 121–128. Springer, 2011. 16

[91] Caroline Rougier, Jean Meunier, Alain St-Arnaud, and Jacqueline Rousseau. Monocular 3d head tracking to detect falls of elderly people. In *Engineering in Medicine and Biology Society. EMBS'06. 28th Annual International Conference of the IEEE*, pages 6384–6387. IEEE, 2006. 10

[92] Caroline Rougier, Jean Meunier, Alain St-Arnaud, and Jacqueline Rousseau. Fall detection from human shape and motion history using video surveillance. In *Advanced Information Networking and Applications Workshops, AINAW'07. 21st International Conference on*, **2**, pages 875–880. IEEE, 2007. 55

[93] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *Pattern Recognition, ICPR . Proceedings of the 17th International Conference on*, **3**, pages 32–36. IEEE, 2004. 26

[94] S. Cheung Sen-Ching and Chandrika Kamath. Robust techniques for background subtraction in urban traffic video. In *Electronic Imaging*, pages 881–892. International Society for Optics and Photonics, 2004. 4, 44, 45, 48, 50, 79, 81, 91, 95, 108

[95] Andre Senior and Sabri Tosunoglu. Hybrid machine vision control. In *The 18th Florida Conference on Recent Advances in Robotics, Gainesville, FL*, 2005. 51, 53, 54, 79

[96] Priyank Shah and Hardik Modi. Comprehensive study and comparative analysis of different types of background sub-traction algorithms. *International Journal of Image, Graphics and Signal Processing (IJIGSP)*, [8]:47–52, 2014. 46, 48

[97] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1010–1019, 2016. 26

[98] Guangyi Shi, Jiye Zhang, Chao Dong, Peng Han, Yufeng Jin, and Jack Wang. Fall detection system based on inertial mems sensors: Analysis design and realization. In *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), IEEE International Conference on*, pages 1834–1839. IEEE, 2015. 11

[99] Anahita Shojaei-Hashemi, Panos Nasiopoulos, James J. Little, and Mahsa T. Pourazad. Video-based human fall detection in smart homes using deep learning. In *Circuits and Systems (ISCAS), 2018 IEEE International Symposium on*, pages 1–5. IEEE, 2018. 26

[100] Jamie Shotton, Ross Girshick, Andrew Fitzgibbon, Toby Sharp, Matthew Cook, Mark Finocchio, Richard Moore, Pushmeet Kohli, Antonio Criminisi, and Alex Kipman. Efficient human pose estimation from single depth images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **35**[12]:2821–2840, 2013. 13

[101] Nishu Singla. Motion detection based on frame difference method. *International Journal of Information Computation Technology*, **4**[15]:1559–1565, 2014. 45

[102] Markus D. Solbach and John K. Tsotsos. Vision-based fallen person detection for the elderly. *CoRR, abs/1707.07608*, 2017. 26

[103] Vianda S. Stel, Johannes H. Smit, Saskia MF Pluijm, and Paul Lips. Consequences of falling in older men and women and risk factors for health service use and functional decline. *Age and Ageing*, **33**[1]:58–65, 2004. 1, 2, 9

[104] Angela Sucerquia, Jos David Lpez, and Jess Francisco Vargas-Bonilla. Sisfall: A fall and movement dataset. *Sensors*, **17**[1]:198, 2017. 23, 37

[105] Nor Surayahani Suriani and Aini Hussain. Sudden fall classification using motion features. In *Signal Processing and its Applications (CSPA), 2012 IEEE 8th International Colloquium on*, pages 519–524. IEEE, 2012. 53

[106] Anna A. Trofimova, Andrea Masciadri, Fabio Veronese, and Fabio Salice. Indoor human detection based on thermal array sensor data and adaptive background estimation. *Journal of Computer and Communications*, **5**[04]:16, 2017. 14

[107] Marcela Vallejo, Claudia V. Isaza, and Jose D. Lopez. Artificial neural networks as an alternative to traditional fall detection methods. In *Engineering in Medicine and Biology Society (EMBC), 35th Annual International Conference of the IEEE*, pages 1648–1651. IEEE, 2013. 16

[108] Bondu Venkateswarlu and GSV Prasad Raju. Performance analysis and validation of clustering algorithms using soft computing techniques. 62, 64

[109] Chia-Chi Wang, Chih-Yen Chiang, Po-Yen Lin, Yi-Chieh Chou, I-Ting Kuo, Chih-Ning Huang, and Chia-Tai Chan. Development of a fall detecting system for the elderly residents. In *Bioinformatics and Biomedical Engineering, ICBBE 2008. The 2nd International Conference on*, pages 1359–1362. IEEE, 2008. 12

[110] Kun Wang, Guitao Cao, Dan Meng, Weiting Chen, and Wenming Cao. Automatic fall detection of human in video using combination of features. In *Bioinformatics and Biomedicine (BIBM), IEEE International Conference on*, pages 1228–1233. IEEE, 2016. 21, 51, 114, 115

[111] Shengke Wang, Long Chen, Zixi Zhou, Xin Sun, and Junyu Dong. Human fall detection in surveillance video based on pcanet. *Multimedia Tools and Applications*, **75**[19]:11603–11613, 2016. 24

[112] Xue Wang, Hong Liu, and Mengyuan Liu. A novel multi-cue integration system for efficient human fall detection. In *Robotics and Biomimetics (ROBIO), IEEE International Conference on*, pages 1319–1324. IEEE, 2016. 1, 2

[113] Yi Wang, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, Yannick Benezeth, and Prakash Ishwar. Cdnet: an expanded change detection benchmark dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 387–394, 2014. 84

[114] Muyun Weng, Guoce Huang, and Xinyu Da. A new interframe difference algorithm for moving target detection. In *Image and Signal Processing (CISP), 2010 3rd International Congress on*, **1**, pages 285–289. IEEE, 2010. 46

[115] D Wiley. *Falls among older adults: An overview*, 2013. 9

[116] Nuttapong Worrakulpanit and Pranchalee Samanpiboon. Human fall detection using standard deviation of c-motion method. *Journal of Automation and Control Engineering*, **2**[4], 2014. 53, 61, 74

[117] Ning Xu and Narendra Ahuja. Object contour tracking using graph cuts based active contours. In *Image Processing. Proceedings. International Conference on*, **3**, pages III–III. IEEE, 2002. 50

[118] S. Yaseen, A. Al-Habaibeh, D. Su, and F. Otham. Real-time crowd density mapping using a novel sensory fusion model of infrared and visual systems. *Safety Science*, **57**:313–325, 2013. 121

[119] Mehran Yazdi and Thierry Bouwmans. New trends on moving object detection in video images captured by a moving camera: A survey. *Computer Science Review*, **28**:157–177, 2018. 45, 46, 47, 48, 49

[120] Miao Yu, Adel Rhuma, Syed Mohsen Naqvi, Liang Wang, and Jonathon Chambers. A posture recognition-based fall detection system for monitoring an elderly person in a smart home environment. **16**, pages 1274–1286, 2012. 56, 59

[121] Miao Yu, Yuanzhang Yu, Adel Rhuma, Syed Mohsen Raza Naqvi, Liang Wang, and Jonathon A. Chambers. An online one class support vector machine-based person-specific fall detection system for monitoring an elderly individual in a room environment. **17**, pages 1002–1014, 2013. 52, 59, 91

[122] Yixiao Yun and Irene Yu-Hua Gu. Human fall detection in videos by fusing statistical features of shape and motion dynamics on riemannian manifolds. *Neurocomputing*, **207**:726–734, 2016. 13, 17, 55, 114, 115, 116

[123] Yixiao Yun, Christopher Innocenti, Gustav Nero, and Irene Yu-Hua Gu. Fall detection in rgb-d videos for elderly care. In *E-health Networking, Application & Services (HealthCom), 17th International Conference on*, pages 422–427. IEEE, 2015. 59

[124] Nabil Zerrouki, Fouzi Harrou, Amrane Houacine, and Ying Sun. Fall detection using supervised machine learning algorithms: A comparative study. In *Modelling, Identification and Control (ICMIC), 8th International Conference on*, pages 665–670. IEEE, 2016. 16, 64, 67

[125] Bing Zhang, I. Horvth, Jonhan FM Molenbroek, and C. Snijders. Using artificial neural networks for human body posture prediction. *International Journal of Industrial Ergonomics*, **40**[4]:414–424, 2010. 1, 2, 9, 10, 11, 12, 65

[126] Chenyang Zhang, Yingli Tian, and Elizabeth Capezuti. Privacy preserving automatic fall detection for elderly using rgbd cameras. In *International Conference on Computers for Handicapped Persons*, pages 625–633. Springer, 2012. 21

[127] Zhimeng Zhang, Xin Ma, Hanbo Wu, and Yibin Li. Fall detection in videos with trajectory-weighted deep-convolutional rank-pooling descriptor. *IEEE Access*, **7**:4135–4144, 2019. 26

[128] Zhongna Zhou, Xi Chen, Yu-Chia Chung, Zhihai He, Tony X. Han, and James M. Keller. Activity analysis, summarization, and visualization for indoor human activity monitoring. *Circuits and Systems for Video Technology, IEEE Transactions on*, **18**[11]:1489–1498, 2008. 12

[129] YOULIAN ZHU AND CHENG HUANG. An improved median filtering algorithm for image noise reduction. *Physics Procedia*, **25**:609–616, 2012. 48

[130] XIAODAN ZHUANG, JING HUANG, GERASIMOS POTAMIANOS, AND MARK HASEGAWA-JOHNSON. Acoustic fall detection using gaussian mixture models and gmm supervectors. 2009. 12

[131] YANIV ZIGEL, DIMA LITVAK, AND ISRAEL GANNOT. A method for automatic fall detection of elderly people using floor vibrations and soundproof of concept on human mimicking doll falls. *Biomedical Engineering, IEEE Transactions on*, **56**[12]:2858–2867, 2009. 10, 12