Development of a Healthcare Software System for the Elderly

PhD Thesis

Laila Ahmad Alhimale

This thesis is submitted in partial fulfillment of the requirements

for the degree of Doctor of Philosophy

Software Technology Research Laboratory

De Montfort University

Leicester - United Kingdom

August 2013

Dedication

”To the pure spirit of my father, who despite his death he is always with me. I wish you were here.”

Abstract

This research focused on the implementation of a reliable intelligent fall detection system so as to reduce accidental falls among the elderly people. A video-based detection system was used because it preserved privacy while monitoring the activities of the senior citizens. Another advantage of the video-based system is that the senior citizens are able to move freely without experiencing any hassles in wearing them as opposed to portable fall detection sensors so that they can have a more independent and happy life. A scientific research method was employed to improve the existing fall detection systems in terms of reliability and accuracy. This thesis consists of four stages where the first stage reviews the literature on the current fall detection systems, the second stage investigates the various algorithms of these existing fall detection systems, the third stage describes the proposed fall detection algorithm in detecting falls using two distinct approaches.

The first approach deals with the use of specific features of the silhouette, an extracted binary map obtained from the subtraction of the foreground from the background, to determine the fall angle (FA), the bounding box (BB) ratio, the Hidden Markov Models (HMM) and the combination of FA, BB, and HMM. The second approach used is the neural network approach which is incorporated in the algorithm to identify a predetermined set of situations such as praying, sitting, standing, bending, kneeling, and lying down. The fourth stage involves the evalua-

tion of the developed video-based fall detection system using different metrics which measure sensitivity (i.e. the capacity of the fall detection system to detect as well as declare a fall) and specificity (i.e. the capacity of the algorithm to detect only falls) of this algorithm. The video camera was properly positioned to avoid any occluding objects and also to cover a certain range of motion of the stunt participants performing the falls. The silhouette is extracted using an approximate median filtering approach and the threshold criteria value of 30 pixels was used. Morphological filtering methods that were dilation and erosion were used to remove any spurious noises from the extracted image prior to subsequent feature analysis. Then, this extracted silhouette was scaled and quantised using 8 bits/pixel and compared to the set of predetermined scenarios using a neural network of perceptrons. This neural network was trained based on various situations and the falls of the participants which represent inputs to the neural network algorithm during the neural learning process.

In this research study, the built neural network consisted of 600 inputs, as well as 10 neurons in the hidden layer together with 7 distinct outputs which represent the set of predefined situations. Furthermore, an alarm generation algorithm was included in the fall detection algorithm such that there were three states that were STATE_NULL (set at 0), STATE_LYING (set at 1) and STATE_ALL_OTHERS (set at 2) and the initial alarm count was set to 90 frames (meaning 3 seconds of recorded consecutive images at 30 frames per second). Therefore, an alarm was generated only when the in-built counter surpassed this threshold of 90 frames to signal that a fall occurred.

Following the evaluation stage, it was found that the combination of the first approach fall detection algorithm method (fall angle, bounding box, and hidden Markov) was 89% with specificity and 84.2% with sensitivity which is better than individual performance. Moreover, it was found that the second approach fall detection algorithm method (neural network performance) 94.3% of the scenarios were successfully classified whereby the specificity of the developed algorithm was determined to be 94.8% and the sensitivity was 93.8% which altogether show a promising overall performance of the fall detection video-based intelligent system. Moreover, the developed fall detection system were tested using two types of handicaps such as limping and stumbling stunt participants to observe how well this detection algorithm can detect falls as in the practical situations encountered or present in elderly people. In these cases it was found that about 90.2% of the falls were detected which showed still that the developed algorithm was quite robust and reliable subjected to these two physical handicaps motion behaviours.

Declaration

I declare that the work described in this thesis is original work undertaken by me for the degree of Doctor of Philosophy, at the software Technology Research Laboratory (STRL), at De Montfort University, United Kingdom.

No part of the material described in this thesis has been submitted for any award of any other degree or qualification in this or any other university or college of advanced education.

This thesis is written by me and produced using LaTeX.

Laila Alhimale

Publications

1. Laila A. Alhimale, Hussein Zedan, and Ali H. Al-Bayatti. Fall Detection system for Elderly people using an Intelligent Video Camera. *In Proceeding of the Fourth IEEE International Symposium on Innovation in Information and Communication Technology (ISIICT).* Amman, Jordan, 2011.

2. Laila A. Alhimale, Hussein Zedan, and Ali H. Al-Bayatti. The Implementation of an Intelligent and Video based fall detection system using Neural Network, Submitted to Elsevier, Applied Soft Computing Journal, 2012.

Acknowledgments

In the name of God. "God raises those who believe and those who were given science degrees." **Great truth of God**. Many thanks to **Allah**, the Lord of the Worlds, if not for whose bounty the completion of this research would not have been possible, and peace and blessings be upon our Prophet **Muhammad**.

Proudly I would like to thank the brilliant scientist **Professor Hussein Zedan** for his encyclopaedic knowledge; he embodies everything that is best in a supervisor. I thank him for his encouragement and the support that he gave me, without which I would have never reached this stage. I am also grateful for his instructive and constructive comments. I am proud and honoured to have been placed under his supervision. Parts of God all the best.

Deep thanks to God, who honoured me with my previous second supervisor, **Dr. Ali Al-Bayatti** who is a symbol of sophistication, humility, tenderness, hope, and optimism. I would like to thank him very much for his valuable guidance and for his continuous support, which has guided the success of my work. Parts of God all the best.

My gratitude also goes to my dear husband **Mahmoud Alazomi** who often encourages me and has supported my morale from this time until the completion of this thesis. I also thank my beloved children very much; they were a source of inspiration for me to continue with this research and also represent my source of energy from which I have summoned the courage to carry on with my project.

A big thanks also goes to the most precious human being in my heart who is my beloved mother, to whom I attribute the greatest credit for the completion of this project. Great thanks and respect also go to my brothers for the big sacrifice they have made for me while I was conducting this research. Moreover, special thanks to my precious spiritual twin sister.

I also seize this opportunity to thank all my friends. In addition, I would like to express my gratitude to my colleagues **Dr. Murad Magableh and Dr. Adeeb Al-Najar** who participated in my research studies assisting me in collecting data for analytical purposes.

Coupled with this, I thank the entire STRL staff team and all my dear colleagues as well as saluting them on their helpful spirit.

Lastly but not least, I am pleased to thank the members of the Committee for participating in the examination of this research thesis; it is an honour for me that my research is being reviewed, examined and discussed by senior professors such as yourselves.

Contents

List of Figures

List of Tables

List of Abbreviation

RGB	Red, Green, Blue
FA	Fall Angle
BB	Bounding Box
HMM	Hidden Markov Model
SE	Structure Element
HHMM	Hierarchical Hidden Markov Model
LHMM	Layered Hidden Markov Model
HPH	Horizontal Projection Histogram
GMM	Gaussian Mixture Model
EM	Expectation Maximisation
MAP	Maximum A Posteriori
SVM	Support Vector Machine
DFT	Discrete Fourier Transform
MLP	Multilayer Perceptrons
QHPH	Quantised Horizontal Projection Histogram

DBN	Dynamic Bayesian Network
ADALINE	Adaptive Linear Element
TP	True Positive
FN	False Negative
TN	True Negative
FP	False Positive
PLA	Perceptron Learning Algorithm
NN	Neural Network
PDA	Personal Digital Assistant

Chapter 1

Introduction

Objectives

- Provide an introduction and state the motivation of this research.
- Define its scope and state the research question
- Outline the research methodology
- Elaborate the thesis structure together with the objectives of each chapter

1.1 Overview

This thesis reports work to develop a novel intelligent fall detection system to minimise serious injuries, specifically to senior citizens, caused by unexpected falls. Often, following such falls, elderly people are left on the ground for some time before the required medical help or support is summoned to them. This unattended time can be critical for some seriously injured older people; therefore, it is increasingly urgent to develop a reliable fall detection system to meet the safety needs of elderly people.

The proposed fall detection system is based on video imaging, whereby video sensors monitor only the silhouette (i.e. an image consisting of black and white pixels) of the person's movements in an indoor environment. In order to preserve privacy, the system does not store the *RGB* image of the person (the *RGB* model is a type of additive colour model whereby red, green and blue light are combined so that an array of colours can be reproduced). The silhouette is scaled and planar quantised, then each frame of the video sequence is compared with a number of predefined situations using a neural network algorithm.

The algorithm recognises the silhouette and the activity recognition is state-based, so that the activity is interpreted for each frame by including information on the timing of each event. The main parameters of the video-based fall detection system are the background modelling and subtraction, the criteria used for the extraction of the person's silhouette and the metrics used to detect falls. A combination of features are extracted from the silhouette, including the fall angle of the person's silhouette, the aspect ratio of the bounding box contouring the silhouette, which relies on the x and y dimensions of this rectangular bounding box, together with hidden Markov models. These features were utilised to improve the robustness of the methods used for detecting falls. Next, a neural network of perceptrons was used to compare the Boolean results of this set of feature-extracting algorithms with predefined scenarios or situations: standing, sitting, kneeling, lying, bending and praying. This thesis describes how the neural network was trained and how activity detection was effectuated. The training of the neural network involved setting initial coefficients which the algorithm modified, improving the system through an iteration algorithm. It was found that the novel detection system produced reliable results in fall identification when compared to other fall detection techniques. Three important characteristics of this fall detection system are promising for its concretisation: its accuracy, its reliability and its protection of the person's privacy.

1.2 Motivation

Given the serious consequences which accidental falls may have for senior citizens, there is a need to find a way to monitor their physical activities so that help or support can be provided to them on time, potentially saving their lives. The current portable sensors or fall detection systems do not seem to be reliable, based on their rates of success in preventing or detecting sudden falls among the elderly and in providing timely assistance to injured persons. However, a review of literature indicates that video-based fall detection systems show some promise and reliability in detecting falls, as they can be installed anywhere at home or in public places and are easily maintained. Therefore, this research seeks to design and assess an intelligent video-based fall detection system which, unlike existing ones, will allow elderly people to move freely and safely while preserving their privacy so that they can enjoy a happy life.

1.3 Scope and Research Question

The proportion of elderly people in the population is growing very quickly; the percentage of those aged over 60 years is predicted to double in the next two decades [8]. Therefore, one of the main objectives of any society is to make sure that proper care is provided to elderly people so that they can live as far as possible healthy, independent and happy lives [9]. However, it is known that unexpected falls, specifically in elderly people, can cause serious injuries with associated medical complications such as internal bleeding and hypothermia, which may eventually lead to fatalities [10].

A fear of falling may curb elderly people's mobility and compels them to live in isolation or in an assisted care environment. One possible solution is to make use of an efficient fall detection system, which can increase their confidence and enable them to continue enjoying their active lifestyle [11]. One advantage of the use of a fall detection system is that it can replace human surveillance by monitoring thoroughly a person's activities through automated technologies. Its sensors can detect warning signs of a fall so that human assistance or medical care can be sent on the spot to reduce aggravation of injuries. There is thus a strong demand for such a systems to reduce the risk of complications arising from injury and the cost of care. The most popular existing method to detect falls is a portable sensor worn or attached to various parts of the human body [12]. However, these sensors have a number of drawbacks: elderly people have a tendency to forget to wear them; they may not feel comfortable in wearing them; and the standard portable sensors do not provide easily interpretable data [13, 14]. Therefore, the aim of this research is to address the following issues, which represent its objectives:

- Firstly the objective is to develop a proper fall detection system which can be accepted and regarded as user-friendly by elderly people, reducing the need for them to live in isolation;

- Secondly, the fall detection system should be able to preserve the person's privacy while tracking meticulously their daily physical activities;
- Thirdly, the system should be highly reliable and accurate in detecting falls so that it can be used confidently to relieve the anxiety and increase the safety of elderly people;
- Fourthly, the fall detection system should be cost effective (in terms of installation and maintenance), adaptable, environment-friendly, quick and easy to install and manageable.

Research question

The aforesaid issues are addressed in this research to answer the following question: "Can more reliable, accurate, user-friendly and cost effective fall detection sensors be devised and built while preserving users' privacy as compared to existing portable fall detection sensors?"

1.4 Research Methodology

The methodology employed is appropriate to scientific research whose aim is to develop and improve existing video-based fall detection technology by including novel modelling methods and techniques and incorporating several engineering principles so that a reliable and accurate system is produced. It was therefore necessary to review the various existing engineering techniques used for fall detection systems [15] as well as the variety of such systems, based on the aims and objectives of this research. The research methodology can be divided into four stages. The first was a review of the literature on existing fall detection systems; the second elaborated the techniques and algorithms used in these systems; the third proposed a novel algorithm to detect and identify falls using two distinct approaches; and the fourth stage was concerned with the evaluation of the proposed algorithm using various metrics and techniques to analyse its sensitivity and specificity in detecting real falls. These four stages are outlined as follows:

stage 1: Research background

The research background was established mainly through a review of the literature on fall detection systems and the hardware and software techniques they have employed, which are associated with the research question. The resources that were used in this secondary research included digital libraries (scientific research papers), books in related fields and articles.

Stage 2: Software model of the fall detection algorithm

The next stage was to investigate the principles behind the implementation of fall detection algorithms, which were then used for the concretisation of a reliable fall detection algorithm. This stage of the research also investigated some engineering techniques that could be included in the fall detection algorithm to increase its reliability and accuracy.

Stage 3: Development of the algorithm

This third stage was concerned mainly with the crystallisation of the fall detection algorithm using programming software tools to meet the objectives of this research. This involved processing silhouettes for comparison purposes and testing the ability of the software algorithm to detect falls.

Stage 4: Evaluation

The fourth stage began with a further review of the literature, this time concerned with existing techniques to evaluate the performance of the fall detection algorithm. The appropriate evaluation metrics were then applied to the empirical data so that the accuracy of the fall detection algorithm and technique developed at the third stage could be compared with that of other fall detection systems.

1.5 Measures of Success

The success criteria for the work reported in this thesis are based on the following factors:

- Firstly, meeting all the objectives stated in section 1.3 above;
- Secondly, a study demonstrating how the proposed fall detection algorithm outperforms the other existing systems;
- Thirdly, a study of why a neural network algorithm was selected among the other methods and included in the fall detection system;
- Fourthly, a study of the various advantages of the novel type of fall detection algorithm compared to existing ones.

1.6 Thesis Structure

Based on the objectives set out above, the following is an outline of the remaining chapters of this thesis, together with a summary of their contents:

Chapter 2: Review of Image Processing and Tracking Methods

The second chapter defines and explains the fundamentals of image processing and reviews various image processing techniques. This involves background modelling and subtraction, where the foreground of the video image is separated from the background to produce a silhouette, i.e. a binary map which distinguishes the person from the background. There is then a review of the literature on the workings of an approximate median filtering method, which is essential in comparing each frame with the background image. A mixture of Gaussians is also compared with the approximate median filtering technique, while techniques such as silhouette extraction are studied together with feature evolution over time for fall detection, the aim being to understand how to automate a fall detection system.

Chapter 3: Review of fall detection systems and methods

The rationale for investigating fall detection methods as well the various types of fall detection methods that currently exist are reviewed in this chapter, which also describes the hardware and software used for the development of fall detection methods as well as their purposes. These methods are reviewed in terms of reliability, fall detection rate, accuracy and practicability. Specifically, there are reviews of two systems, using portable sensors and video cameras respectively. There is also an elaboration of the methods used to detect falls including the fall angle, the bounding box method and the hidden Markov models. The chapter finally reviews the neural network algorithm to be employed in the implementation of the novel fall detection system, examining all previous studies exploiting this method of fall detection and comparing these to the methods used in the present research.

Chapter 4: Mathematical concepts

The fourth chapter examines all the mathematical concepts used in the concretisation of the novel fall detection algorithm, covering image processing (background modelling and subtraction), silhouette extraction methods and fall detection methods: fall angle *(FA)*, bounding box *(BB)* and hidden Markov models *(HMMs)*. The FA represents the angle between the ground and the person's centroid; the *BB* method represents a rectangular box that surrounds the moving object and whose dimensions on the horizontal and vertical axes are used to evaluate the aspect ratio; the *HMMs* are based on probability theory and this chapter also describes the transition and emission matrices that are found by the *HMM* train function. There is a full explanation of the neural network and respective algorithms and of how the neural network algorithm was used to identify activities and to classify falls and non-falls.

Chapter 5: Neural network

This chapter presents the rationale for using a neural network and explains how it forms part of an intelligent surveillance video system. It describes the output sequence and its corresponding predetermined set of situations such as blank screen, standing, sitting, lying, kneeling, bending and praying. It also explains the importance and various uses of neural networks and describes the underlying principles of the neural network used, together with its topologies. The working principle of a neurone and the activation functions are briefly described. There is a discussion of the learning algorithm of the back propagation method and the training procedure involved and the chapter ends by explaining the principle of genetic algorithms.

Chapter 6: Preprocessing techniques

The sixth chapter sets out the techniques utilised for the preprocessing of the images comprised of background modelling and subtraction (by distinguishing between foreground and background pixels) through the use of an approximate median filtering method and the conversion of *RGB* images to greyscale. It then shows how the silhouette is extracted, i.e. specific important features are removed for analysis purposes. These extracted features are subsequently used for detecting falls.

Chapter 7: First approach to fall detection

This chapter describes the study of feature evolution with time and the detection of a fall. It explains the first approach used for the fall detection algorithm, employing the *FA*, *BB* and *HMM* methods individually and in combination, as they make use of specific features of the video images in the development of the fall detection algorithm. The limitations of these methods are explained, as are the ways in which neural networks can be used to improve the performance of the fall detection algorithm. The experimental results of this first approach are presented in this chapter.

Chapter 8: Second approach, using a neural network

The second approach, using a neural network algorithm, extends the theory and algorithm of the existing fall detection system by identifying all the situations and by distinguishing falls from various other situations such as sitting, praying and standing. This chapter elaborates the situation identification after the silhouette has been extracted and normalised. It also describes two specific problems, the first related to the position of the silhouette, which can appear anywhere in the picture frame, and the second concerning the unknown size of the silhouette. A neural network of perceptrons was used to classify the silhouette according to a set of predefined situations: standing, sitting, bending, kneeling, lying and praying. There is a full account of the development of this neural network (consisting of a hidden layer, number of neurons and output vector) as well as how the input data were used for the neural network. The most important part of this chapter concerns the training of the neural network using supervised learning. It also explains how the frames were selected to train the neural network and describes the MATLAB functions used in the algorithms for the development of the fall detection algorithm. The results of the identification of situations using the neural network are presented for praying and different types of fall. Finally, the state machine is depicted and explained to show the working of the algorithm and the states through time.

Chapter 9: Evaluation

This chapter describes the methods used to evaluate the fall detection system implemented in this research by comparing it to other video-based and portable fall detection sensors. Then the neural network algorithm is evaluated in terms of accuracy and of convergence of the solutions for the weights of the neural network. The fall detection system is assessed in terms of its ability to distinguish a fall from a non-fall scenario and the algorithm codes are evaluated in terms of computational time and complexity, as well as the setting of the parameters for the alarm system to be unbiased and reliable. This chapter also deals with the evaluation of the first approach to detecting falls, involving *FA*, *BB* and *HMMs* and a combination of these methods, and of the second approach, involving the neural network algorithm. It explains the implementation of evaluation metrics such as sensitivity and specificity in order to compare mathematically the accuracy and reliability of the fall detection algorithm and the use of simulated limping and stumbling by 'stunt participants'. Finally, the limitations and drawbacks of the intelligent video-based fall detection system are discussed.

Chapter 10: Case Studies

The case studies described here involve datasets related to specific activities such as falling, sitting and praying. For example, the dataset fall5 represents a specific type of fall where the person experiences a lateral rotation of the knees. The cases studied include other types of fall such as backward, sideways and forward falls, as well as neutral activities such as sitting on a chair, kneeling, squatting, standing still, walking and praying. A final case study demonstrates brief falls followed by standing up.

Chapter 11 Conclusion and Future Work

The final chapter summarises the findings of the novel fall detection algorithm and its performance as compared to other existing methods. It also emphasizes the importance of the implementation of an intelligent video-based fall detection system which is reliable, secure and maintains a level of privacy while producing consistently accurate fall detection results. Recommendations are made to consolidate the current work by means of future research to improve the fall detection system developed in this research.

1.7 Original Contributions

The intelligent fall detection system which was devised in this research shows promising facets in terms of very high reliability and also accuracy. The fall detection system detects in average about 94.3% of the 420 tests of falls or no falls for 10 participants which obviously improve the existing research works of Homa [16] whereby in her fall detection model, only 89.5% of the falls or no falls were detected and also she used participants of similar age group. In addition, she did not include daily Muslim activities such as praying for not be confounded with real falls. Therefore, in my research I have included praying so that the fall detection can distinguish between fall and praying and also the type of participants used include those age group of people analysed in (Homa and others') studies. Also, I have added a variety of people of different age group to challenge the fall detection system in detecting falls despite the different body size or physical attributes.

Chapter 2

Image processing and tracking methods

Objectives

- Examine the importance of camera systems and difficulties in implementation
- Provide a review of image processing techniques
- Discuss image filtering methods
- Elaborate the tracking features that can be used to automate a fall detection system

2.1 Introduction

Elderly people easily forget to wear the common portable fall detection sensors, or sometimes they feel irritated at having to take these with them wherever they go [17]. These drawbacks prompted this research to investigate the promising video-based fall detection method, which preserves privacy and gives users more freedom in their daily physical activities. However, in order for video surveillance systems to be efficient and accurate, they need to be robust regarding image processing difficulties. The first one is the type of camera to be used in the video-based fall detection system. With cheap cameras, the video sequences will be highly compressed, as is the case for MPEG-4, creating artefacts in the image. Moreover, there may be a variable illumination that can be observed, which it is important to take into consideration, specifically while updating the background process. Another issue to address with a camera system is that of privacy. The senior citizens should be aware that they are being filmed, even though nobody will view the video; hence it is important to have permission to install these cameras in their homes, for instance. Furthermore, the users should be informed that the video data will be processed only by a computer, as they may be filmed in private situations; for example, bathrooms are high fall-risk locations. Therefore, it is important to process the original image to a binary one, so as to address these privacy concerns.

Lighting can represent another issue, by producing reflections in the scene (sometimes the colours can be become brighter than usual) or shadows from the moving body (i.e. the colours become darker than usual). Such reflections and shadows from moving objects can lead to detection errors. Another source of error can be occlusion, which can result because of the presence of furniture (chairs, sofas) or doors in the field of view. Objects being carried, such as bags or clothes, can also produce occlusions, while moving objects that are not of interest for fall detection can cause 'phantoms' in the image and must be dealt with as part of the background. A robust fall detection system using video cameras should not generate unnecessary alarms because of these image processing issues. Indeed, some precautions can be taken to reduce them. A good inexpensive camera should be chosen and placed high in the room to reduce occlusion by objects and to allow a large viewing field. In this research, a video of 300x450 pixels with compression Indeo5 and video frame rate 30 fps was used. The camera was placed nearly perpendicular to the direction of motion of the participants at a distance of about 2.5 metres, so that the whole body of each participant was captured during motion.

Image processing techniques used in this research

This section outlines the image processing techniques that were employed for the concretisation of the video-based fall detection system. First, the video sensor was positioned in a confined room or laboratory so that it could capture three to four strides of a walking person and the initiation of a fall or the actions of sitting down or praying; it also had to cover the full height of each participant while moving in its range. As far as possible, there were no occluding objects in front of the camera which could impair its view or the recorded scenes of the moving person, so that occlusions could be reduced for analysis purposes. At the initial stages of image processing, the recorded *RGB* images were converted to manageable greyscale images to facilitate analysis and data handling, using an RGB-to-greyscale algorithm which takes as input the pixel values of the three primary colours: red, green and blue.

The background subtraction method utilised in this research was based on approximated median filtering because it is simpler than other existing models (discussed in section 2.3). In addition, this subtraction method represents a fast algorithm requiring minimal processing power and gives good results compared to other methods [18, 19, 20, 21]. From the background modelling, the silhouette of the moving person was extracted. In fact, the silhouette represents a binary map which discerns the individual outline from the background image so that it can be used for further analysis, preserving the privacy of the person being recorded. Afterwards, morphological filtering techniques involving dilation and erosion were used to clean the silhouette of any unwanted or spurious noise, which can pose serious problems in the silhouette image and hence in the analysis and in the development of a successful video-based fall detection system.

Once the silhouette has been extracted and processed, it can be tracked so that the behaviour of the moving body can be monitored and its position recorded. The tracking features can include the shape of the tracked body (in terms of height or width of the silhouette) and for this research, the spatial features of the silhouette were exploited for tracking. The challenges that were encountered in this work were in ensuring that the tracking of the moving body was performed correctly and in deciding the type of spatial features to be used for this purpose.

2.2 Methods of image processing

The video input colour images were converted to greyscale because of the following crucial advantages:

- Achieving a reduction in processing power and memory consumption.
- Sending greyscale video data over a network allows higher resolution than using colour video data [19].

This section reviews the techniques used in processing the video images for detecting falls. First of all, it is important to know that an image is an array of pixels that are arranged in columns and rows [20, 21]. For example, in an 8-bit greyscale image, each pixel of the image has an intensity which can range from 0 to 255. A greyscale image is composed of black, white and different shades of grey (Figure 2.1). The RGB colour model is an additive colour model in which the colours red, green and blue light are added together to produce a broad array of colours [22]. A video camera is an example of a typical RGB input device.

Figure 2.1: A traditional greyscale image of pixels displayed in rows and columns [1]

Morphological filtering

The morphology of an object refers to its form or structure, while morphological filtering simplifies a segmented image so as to probe the objects of interest more easily. This probing process can be performed by smoothing out the outlines of objects, filling small holes, deleting small projections or other related techniques.

There are two principal morphological operations: dilation and erosion. Dilation consists of expanding objects of interest by eroding their boundaries. This is done by placing a structuring element (SE) on the image and sliding it across it [23].

Morphological image processing is extensively used to process binary and greyscale images. An image is regarded as a subset of a Euclidean space R^d or the integer grid Z^d in some dimension d in binary morphology. A Euclidean space is a set of points which satisfy certain associations that can be formulated in terms of angle and distance [24]. The main functions of binary morphology are to search an image with a predefined shape and to compare how this shape fits the shapes in the image. This probe is sometimes called the structuring element and is a subset of the space or the grid [25, 26].

The two basic operations of dilation and erosion can be blended to form complex sequences. The two most important types of morphological filtering are opening and closing. The opening process, which consists of an erosion followed by a dilation, is utilised to remove pixels from regions that are too small to cover the structuring element [27]. This structuring element's task is to search the image for small objects to filter out. An example of opening is shown in (Figure 2.2).

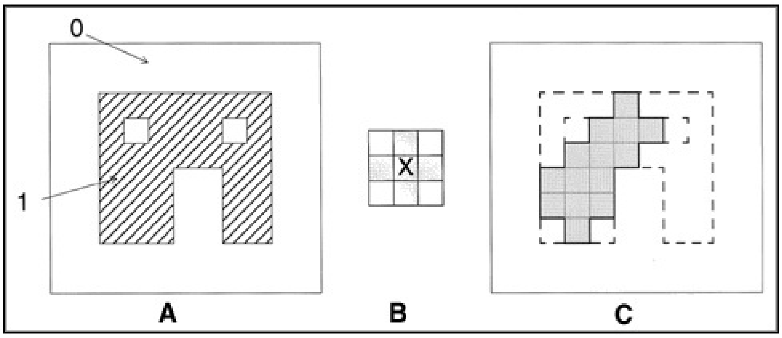


Figure 2.2: Opening process: A is the original image, B is the structuring element; X is the origin of the structural element, C is the image after opening, i.e. erosion followed by dilation [2]

The closing technique, which consists of dilation followed by erosion, can be used to fill in holes and close small gaps, as shown in (Figure 2.3). The order of operation is crucial for the desirable result to be obtained during morphological filtering.

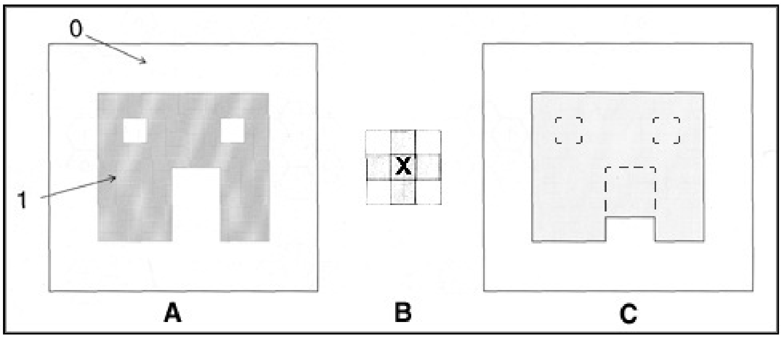


Figure 2.3: Closing process: A represents the original image; B is the structuring element with X as its origin; C is the image after closing, i.e. dilation followed by erosion; original in dashes [2]

2.3 Background modelling

This section describes the background subtraction algorithms which can detect moving objects. As seen in (Figure 2.4), background subtraction immediately follows the video capturing stage. Its objective is to extract the person's silhouette from the video input by subtracting a background estimation model from this original video input. This step allows the fall detection algorithm to determine features of the person that would seem important for fall detection stage. Before the estimated background can be subtracted from the current frame, the background image needs to be generated from the background modelling stage, which represents a very important step for the whole background subtraction algorithm. The following subsections describe the most practical techniques employed in background modelling, which fall into two main categories: recursive and non-recursive methods.

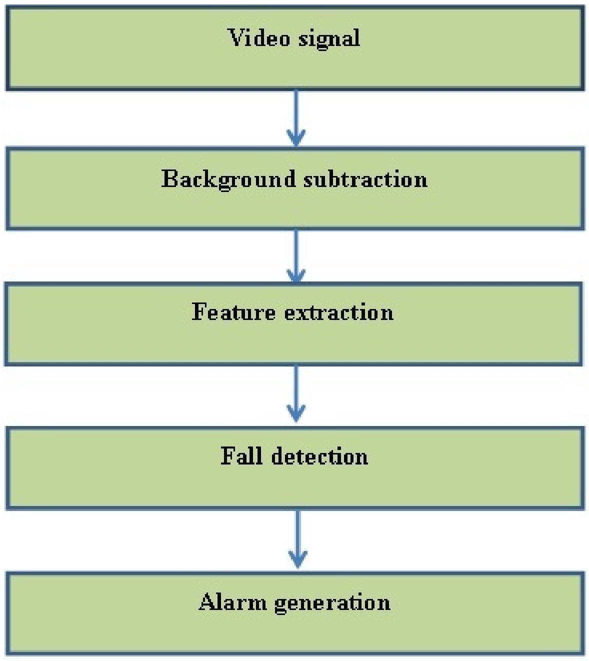


Figure 2.4: A simplified version of the camera-based fall detection system, depicting the image processing stage.

2.3.1 Non-recursive methods

Non-recursive methods utilise a sliding window function, which keeps a certain number (N) of frames in a buffer; based on these buffered frames, an estimation for the background model is effectuated. This technique is adaptive, as the model is determined by the previous N frames and is not influenced by the frames that appeared before these buffered ones. However, this buffer function needs a large amount of memory, especially if a large buffer is utilised [28]. The following three subsections describe the various non-recursive techniques.

2.3.1.1 Frame differencing

Frame differencing is a very simple background modelling method which considers the frame at time *t-1* as the background model. This implies that the background is modelled on the previous frame. In order to obtain the foreground or the moving object, the background model is subtracted from the actual image using some threshold value [29]. This method is sensitive to the threshold value chosen to distinguish the foreground from the background, which is the only factor that can affect the result. Its advantages are low computational load and memory requirements, but it does not provide precise results and is sensitive to noise. This represents a major issue, especially when the person does not move for a brief period of time (more specifically for one frame), because the system will then interpret the person as part of the background.

2.3.1.2 Median filtering technique

Another effective background modelling technique is median filtering [30]. In this background subtraction approach, the background value for each pixel is determined as the median of that pixel in all the frames found in the buffer. Although this technique can give good results while employing low computational power, it does require a certain amount of memory (N x framesize) [29].

2.3.2 Linear predictive filter technique

In the third non-recursive approach, the current background model estimate is determined by applying a linear predictive filter to the pixels of the buffered frames.

In this model, the filter coefficients are calculated from the sample covariances for each frame time [28]. This technique is not reliable in real-time systems because of its lengthy computations.

2.3.3 Recursive methods

The main difference between non-recursive and recursive methods is that the latter do not employ a buffering system consisting of previous frames, but instead update the background image in a recursive manner. The advantage of using such technique is that there is only one frame which is stored and this frame or image is updated whenever a new frame is received [31]. However, when the background image becomes faulty due to some erroneous factor, this fault will take a long time to disappear from the estimated background image. Therefore, recursive methods are less adaptive than non-recursive ones. The following subsections describe the various recursive techniques.

2.3.3.1 Running average method

The running average recursive technique is a simple and fast modelling algorithm which does not require high memory. However, this method does not give accurate results and this normally depends on the type of application (where alpha can be fine-tuned so that it can be acceptable). Overall, it has simple computation and memory requirements, at the expense of poor accuracy [28].

2.3.3.2 Approximated median filtering technique

The approximated median filtering algorithm was developed by McFarlane and Schofield to track animal motion as part of a background modelling method [32]. When a pixel in the current frame has a greyscale value greater than the equivalent pixel in the background estimate, then the pixel value in the estimate is incremented by one. Conversely, when the pixel value in the current frame has a value lower than the pixel value of the background estimate, then the latter is decremented by one. Therefore, when this function is applied to a background model, it converges to an estimate where half of the input pixels are greater than the background model while the other half are less. Approximated median filtering can achieve good results with percentage accuracy close to that of algorithms requiring higher complexity. The amount of memory space required is also low and the technique is robust. Its only drawback is its slow adaptation to major changes in the real background [28].

2.3.3.3 Kalman filtering

This background modelling technique relies on the assumption that the best information on a system state is obtained through estimation [33]. The literature describes several approaches to making this type of estimation [28], most of which use the luminance, the luminous intensity together with its temporal derivatives or luminous intensity with its spatial derivatives. The advantage of this method is that the gain matrix can change from fast to slow adaptation according to whether a pixel is in the background or foreground. The disadvantage of Kalman filtering is that it leaves long trails behind the moving body or object.

2.3.3.4 Mixture of Gaussians

Finally, the mixture of Gaussians is a popular method of background modelling. Unlike the Kalman filtering method, which can track only one Gaussian, this technique can usually track three to five Gaussian distributions simultaneously [34, 35, 36, 37]. Its advantages are that it normally results in good accuracy and needs a fairly modest amount of memory space. A distinct difference from other methods is that the mixture of Gaussians does not utilise one image of values as its background model; rather, each pixel is modelled by a number of Gaussians, which represent the probability distribution function F. While this technique produces accurate results, its disadvantages are complexity of computation and high sensitivity to abrupt changes in illumination.

2.4 Tracking of moving objects

The tracking of moving objects or bodies in a video sequence is the process of finding the same objects or bodies in different images or frames [16]. In order to track these objects, information of their position, *size*, *colour distribution*, *speed*, *shape* and *trajectory* is used. The features based on this information are first extracted or determined, based on the variable values. The main objectives of the object tracking step are to find out when a new body or object enters the scene (the viewing field of the camera system) and then to track it. The tracking step also needs to compute the correspondence between the foreground regions in the current frame as well as the objects currently being tracked. Next, the tracking stage uses tracking algorithms to analyse what the moving objects or bodies are doing and to determine their position in the scene. At this tracking stage, segmentation also needs to be improved by associating the blobs that belong to the same body, also called region merging. The main goal of detecting moving objects in a video stream is to enable the tracking of the objects of interest through time so that a set of properties from the trajectory can be derived, such as the physical behaviours of the physical body [38, 39]. Some common approaches to tracking are token-based (when the geometric object description is available) or intensity-based (correlation or optical flow). However, these techniques are not suitable for tracking blobs, as a reliable geometric description of the blobs cannot be inferred. Intensity-based techniques have a tendency to ignore the geometric description of blobs [40].

In fact, various methods are used to detect moving bodies in video surveillance. For example, [41] describe an object tracking algorithm using object trajectories, greyscale distribution, sizes and texture. These authors assume that the object's acceleration is uniform in a couple of adjacent frames. [42] propose a multi-hypothesis tracking algorithm method for tracing salient moving objects. In this method, filtering and pruning are applied at various levels of processing to remove unwanted objects or trajectories from the tracker. This technique uses extracted features from the foreground, Kalman filtering and colour similarity to solve occlusions. [41] also employ the Kalman filter to predict the movement parameters in the tracking module and develop a tracking matrix to determine whether objects cause occlusion. Other tracking algorithms use a maximum likelihood classification method [43] or a continuously adaptive mean shift algorithm [44]. The tracking stage is extremely important for any type of event detection system. One result of unsuccessful tracking can be a failure of the event classifier.

2.5 Feature extraction and segmentation

Normally, after detecting moving objects and tracking them, the system should extract some important features for event detection from the related data. Each moving object can be defined by a set of features which depend only on the current foreground frame, comprising centroids, heights, bounding boxes, colour histograms and areas [39, 45]. However, other extracted features, such as speed and displacement, can rely on the sequence of frames. These can be classified as spatial features of the objects (width, height) and motion features (speed and movement). The key challenge is to determine the relative features needed to identify the activities occurring in the recording scene. Moreover, these relative features should be quantifiable and different for different activities, so that the event can be classified as the next step or stage. It is expected that this classifier will incorporate the features of vector or scalar values as input and group the activities of the moving body in the scene based on the distinct features of the vector values. Finally, the output of this classifier should have a meaning and be associated with human activities such as falling, praying, sitting or walking.

Image segmentation is the process of dividing a digital image into different segments or sets of pixels so that it is simplified into something easy to interpret and analyse [46, 47]. In addition, image segmentation is commonly used to locate objects or bodies as well as boundaries such as curves or lines in images. The result of image segmentation is a set of segments which together cover the whole image or a set of contours extracted from the image, such as for edge detection.

2.6 Summary

This chapter has defined and described all the various stages of image processing, beginning with the importance of transferring the image from RGB to greyscale. Next, there was a discussion of the background subtraction algorithm methods most commonly used in video-based fall detection systems. Each model was seen to have advantages and disadvantages. The more complex models tend to give better results based on accuracy and robustness, whereas the simpler ones need less processing power and can give satisfactory results. In the present research, approximated median filtering is used to ensure fast processing. The chapter next examined the algorithms used to remove noise from extracted silhouettes, then described the current tracking and feature extraction methods, including image segmentation, as they represent the steps taken before moving to the fall detection stage, dealt with in chapter three. The challenges associated with each discrete sub-stage were emphasized and it was stated how these were overcome in the present research.

Chapter 3

Review of fall detection systems and the underlying working mechanisms

Objectives

- Present the different types of fall detection systems
- Explain the development of fall detection methods
- Describe the existing methods employed to detect falls
- Examine the importance of neural networks in identifying falls
- Present the fall detection methods used in the current research

3.1 Introduction

As healthcare technology progresses, simple devices can be produced to detect and predict falls, especially for elderly people, which could eventually save their lives without intruding on their independence. A fall occurs when someone comes down freely under the influence of acceleration, of free fall or of gravity [48]. The characteristics of fall are different from walking or standing. A fall is a process that lasts from one to two seconds and involves several sub-actions; if the person is standing at the start of a fall, in this case the fall is the transition from standing to lying on the ground. The head is on the floor at the end of the falling process and will remain on the floor with little or no motion during a certain period of time. A person falls approximately in one direction and therefore both the head and the centre of gravity of the person move about one plane during the fall. The height of the head is reduced from that of the standing position to that of the floor and within this period of time, the head will fall freely [49]. The lying head is within a circle centred on the position of the feet at the last standing time and with the radius of the height of the person.

A considerable amount of research has been conducted into fall prediction and fall detection. Fall prediction research is concerned with forecasting the occurrence of a real fall, while fall detection research addresses ways to identify real falls. Work on fall detection is important because its findings can be used to reduce injury or mortality resulting from unexpected falls. Fall detectors should be able to be located in any places where there is a high risk of falls or where falls often occur, such as on slippery floors or surfaces [50, 51]. The main challenges in developing fall detectors are to maximise ease of use, installation, reliability and accuracy while safeguarding the privacy of any person subjected to monitoring.

Falls represent one of the causes of severe or fatal injuries among senior citizens and the risk is increased if the person cannot call for help. As explained in the introductory chapter, mobility is currently regarded as an important issue in sustaining the independence of elderly people. Falls not only represent a major health risk but can greatly affect their mobility; indeed, even the fear of falling can result in decreased mobility and activity [52, 53, 54]. This is why researchers are developing methods to successfully detect and monitor falls to meet the needs of elderly people. This chapter describes the common types of real-time fall detection systems and the methods or algorithms underlying their working principles. These fall detection systems have been regarded as effective by other researchers working to produce an unobtrusive and safe monitoring system for independent senior citizens.

3.1.1 Sensor based fall detection

A review of relevant publications indicates that fall detection can be either sensor-based or video-based. Some researchers in this field have attempted to detect falls using different types of sensors ranging from accelerometers (i.e. devices that detect the direction and magnitude of the acceleration along a single axis or multiple axes), microphones, cameras, gyroscopes (i.e. devices measuring the orientation of a body along one axis or multiple axes), or a combination of these [55, 56]. A sensor-based system normally involves a worn or embedded device which is used to detect falling events. If the wearer of such a device falls, a signal is sent to a response centre for analysis and processing [57, 58]. Video-based systems, for their part, use a video surveillance mechanism and digital processing of real-time and recorded video images to detect whether a falling event has happened [59].

3.1.2 Camera based fall detection

Cameras are normally used in in-home assistive systems, because they are better than sensors and have been made affordable by the decreasing cost of cameras [60]. First, the camera-based approach can detect multiple events simultaneously. Secondly, as they are attached to buildings and not worn by users, they are more user-friendly and less intrusive. Finally, the recorded video images can be used remotely for analysis or verification purposes. Camera-based methods can be categorised into three types, based on some of the major characteristics of falls, which the algorithms used take account of: inactivity detection, 2D body-shape change analysis and 3D head motion analysis.

Camera based methods

Inactivity detection algorithms recognise that a fall will end with a period of inactivity on the floor, which is related to the characteristics of fall from walking or standing. Nait-Charif and McKenna used an overhead omni-camera in their system [59]. The algorithm tracked the person to obtain traces of his or her motion. Jansen and Deklerck utilised a stereo camera to acquire a 3D image on which the body area was identified and its orientation determined [61]. Finally, the lack of change of body orientation was used to detect inactivity and a fall was detected if inactivity occurred in a certain context.

The principle of shape change analysis algorithms is that the shape of a falling person will change from standing to lying and this is related to differences in the characteristics of a fall from walking to standing. These algorithms are using the first characteristics of falls implicitly, owing to the use of *HMMs* that involve a time constraint. An HMM- based fall detection algorithm was proposed by Toeyin where the HMM uses video features to distinguish fall from walking [62]. The features are wavelet coefficients of the height-to-width ratio of the bounding box of body shape. Another type of *HMM* employs an audio feature to distinguish the sound of falling from talking. Anderson used an HMM-based algorithm to detect falls [63]. Their *HMMs* incorporated various features extracted from the silhouette: BB height, motion vector magnitude, covariance matrix determinant and width-to-height ratio of the person's BB. The *HMMs* were trained to distinguish walking, kneeling, getting up and falling. Thome and Miguet used an algorithm based on a hierarchical hidden Markov (*HHMM*) to detect falls [64]. The single feature of the *HHMM* was the orientation of the body's blob, while its state level was the body posture. The other two levels of the *HHMM* represented behaviour pattern and global motion pattern respectively. Miaou used a rule-based algorithm to detect falls [65]. The rules inferred the occurrence of a fall based on the ratio of width to height of the BB of the body in the image. The system used an omni-camera and relied on context information to decide if a fall had occurred.

Cucchiara used multiple cameras, calibrated in advance, to obtain the 3D shape of the body and so to detect falls [30]. Hsu used deformable triangulations of body shape to classify postures (one of which denoted a fall), again obtaining the shape of the body from 3D images [66].A smart camera network that consists of a number of low resolution-cameras was developed by Williams [67]. Although the system had multiple cameras, it was not intended to track people but to detect the result of falls; thus, all cameras simultaneously took images at a very low frame rate. Moreover, a fall detection algorithm based on the 2D shape of the human body extracted from compressed domains; that is, without decompressing the video was developed by Lin [68].

In 3D head motion analysis algorithms, the principle that vertical motion is faster than horizontal motion in a fall was used. An approach to detect falls using monocular 3D head tracking was presented by Rougier, where the tracking component locates the head, estimates the head pose using particle filters, then obtains the 3D position of head [69]. The fall detection component computes the vertical and horizontal velocity of the head and uses two appropriate thresholds to distinguish falling from walking.

The algorithms used in inactivity detection involve a light computing load and can be used in small devices. However, they are more likely to produce false alarm, even where context information is used. They tend to sound the alarm late, because they detect a fall only when the person has been lying on the ground for a while. Comparing shape detection algorithms with head detection is that shape detection algorithm are much more reliable than head detection ones, because the human body is a salient object; and generally when body shape is obtained, fall detection has light computing needs, because the computational load of the algorithms relies on the type of shape detection method which is utilised. Body shape detection can be done in real time, whereas 3D body shape detection needs heavier computation or more cameras and is less reliable. However, the existing algorithms in this category determine too few features from shape motion-mainly the BB width-to-height ratio-and they do not detect sub-actions of fall, except that *HHMMs* may find the posture of the body in its middle layer.

In 3D head motion analysis algorithms, the advantages are that the head has less occlusion and the head motion has higher correlation with fall than body motion. But tracking the 3D position of the head using a single camera is unreliable and time-consuming. One of the main drawbacks of any camera system is the associated privacy concerns [67]. Elderly subjects should be informed that they are being filmed, although nobody will look at the video, and it is crucial to have their permission to install cameras in their home.

The elderly should be aware that the video data is not viewed by other persons but is only processed on a computer, and that the data will not be recorded or used for any purpose, especially as the person may be filmed in a private environment such as the bathroom, which is unfortunately a high fall-risk location. With respect to these privacy concerns, there may be the possibility of sending an alarm message to a remote appliance such as a mobile phone or Personal Digital Assistant *PDA*, possibly including an image of the fallen person. It is also essential to process the original image to a binary one before sending it to a remote computer for processing purposes. Besides the privacy issues, there are other technical challenges that need to be dealt with while implementing a video-based fall detection algorithm.

3.1.3 The main issue associated with fall detection

The main issue for any fall detection system is to identify a fall among all the everyday activities such as crouching, sitting or praying, which bear similarities to falls. Noury posit that a fall event can be dissociated into four specific phases [70], as depicted in (Figure 3.1).

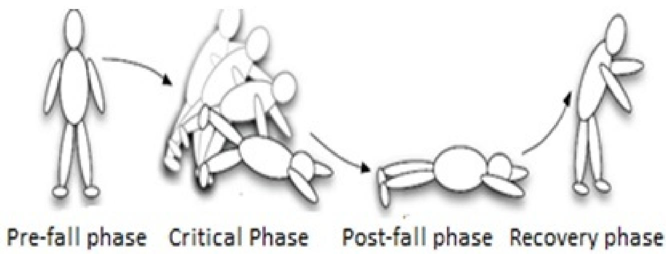


Figure 3.1: The four phases of a fall event [3]

First of all, the pre-fall phase represents normal daily activities which include sudden movements towards the ground or floor, such as sitting, crouching or praying. These activities should not induce any alarm from the fall detection system. Secondly, the critical phase, normally brief, represents the movement of the body towards the ground or the shock of impact of the body with the ground. Thirdly, the post-fall phase is normally characterised by a person staying motionless on the ground just after the fall has occurred. This post-fall phase can be represented by a lying position or an absence of significant motion. Lastly, the recovery phase is the period when the person is able to stand up by himself or through the help of another person.

3.1.4 Alternatives to video surveillance for fall detection

Wearable sensor devices are normally based on accelerometers [71, 72] which can detect both the direction and magnitude of acceleration. Alternatively, gyroscopes are used to measure the orientation of the body of the person [73]. Nyan utilised a combination of accelerometer and gyroscope to detect falls [74]. The limitations of these technologies are that such sensors are usually embarrassing to wear and that the batteries which are required for them to function will need to be recharged often or replaced periodically. Given these drawbacks to sensors, video surveillance may offer an interesting and promising solution for detecting falls, as no body-worn devices are required. Alwan developed another alternative, a floor vibration-based fall detector, but its performance depended on the floor dynamics [75]. This approach was improved by Zigel through the inclusion of a sound sensor [9], which increased the fall detection rate, but the authors acknowledge that low-impact human falls (hence producing sound of low amplitude) may not be successfully detected.

3.1.5 Specific issues with video-based surveillance

A good and reliable surveillance video system is expected to be robust in response to image processing difficulties. Therefore, the correct choice of camera, its position and an appropriate video compression so that artefacts can be reduced are important decisions to make for the implementation of a video surveillance system [3]. Londei found that in a group of 30 senior citizens experiencing sudden falls, about 87% were in favour of this type of intelligent video-monitoring system [76]. The advantages of such video surveillance systems are that they offer a quick and secure intervention for the senior citizens and that video images recorded before fall occurrences can provide important information to better understand the origins of falls. Thus, both security and interventions for fall events can be improved. According to Rougier [3] video surveillance seems to be a promising fall detection system to help elderly people; however, some of them are concerned about the safety and privacy of the transmission of the video images. Furthermore, current research should also improve on the accuracy of the video-based detector, because some studies (e.g. Willems [77]) have shown that detection up to 85% (by recording video images from the side view of the person in motion) and 78% (by recording the images in the frontal view of the moving person) can be achieved using video-based systems.

3.2 Real-time fall detection systems

3.2.1 Acceleration-only sensors

Boyle and Karunanithi developed a movement classification algorithm based upon the acceleration data generated from two axis accelerometers [78]. A healthy young volunteer performed simulated falls close to the recordings of different types of falls experienced by elderly patients. In this study, some types of fall were more easily detected than others (such as falling forward). It was found that the magnitude of the acceleration alone was not reliable. Only when a combination of the magnitude of the acceleration and the speed of change of sign was used did this fall detection algorithm become reliable.

Lindemann used a fall detection system that encompassed two accelerometers placed inside a hearing aid housing device to detect seven types of falling and five types of daily activities [13]. Chia-Chi Wang [79] later extrapolated the method of Lindemann [13] by determining two parameters: the sum of three axial accelerometers (in three dimensions, representing the x, y and z-directions), denoted by Sa, and the sum of the frontal and sagittal acceleration components, denoted by Sh. These two parameters were used to search for significant points of time in the falling process, to determine the velocity of impact of the subject (Vmax) and to identify the lying condition.

Alan and Bourke report that there are a number of fall detection systems based on accelerometers, gyroscopes or optical motion capture systems [80]. These use the velocity profiles of the trunk of the human body for the detection and prediction of falls. An accelerometer-based sensor worn on the chest was used for fall detection.

3.2.2 Detection using a combination of sensors

Narayanan developed a waist-mounted rechargeable triaxial accelerometer for the detection and prevention of falls by senior citizens [81]. This device serves two purposes: for monitoring and to administer fall risk assessment to elderly people. These fall risk assessments can include timing sitting/standing repetitions and measuring reaction times, which are effective parameters for evaluation purposes. Whilst the device was tested on only one individual, its ability to detect falls autonomously and to allow the person to provide input and self-testing results appears to offer a simple but robust fall detection and prevention method.

A study by Noury shows that the worst-case scenario of the psychological consequences of a fall is that the senior citizen becomes unconscious following a fall and is therefore unable to summon help. Noury reassures elderly people that help will be brought to them shortly after a fall based on the fall detection system he has designed [82]. The device uses three sensors to create an 'actimeter', which monitors the orientation of the human body, taking into account the surface vibrations and the vertical acceleration using a position tilt switch, a vibration sensor and an accelerometer respectively. The summed outputs of the three sensors produce a Boolean value that is subsequently used to determine whether or not a person has fallen. **Non-Acceleration methods**

Popescu and colleagues developed a fall detection method consisting of an array of acoustic sensors [83]. Both the amplitude of the sound (for example, a loud sound indicates that a fall has occurred) and its location (whether high or close to the ground) were determined. Form a combination of these two parameters, an attempt was made to determine whether a fall had in fact occurred. The authors state that the device was able to detect only 70% of falls. They succeeded in adjusting it to detect 100% of falls, but it then gave five false alarms per hour.

Ng used a different technique to track mobility using a monocular vision system. The device was first fixed to the elderly person's walker [84]. Its camera detected the positioning of his or her legs for several frames to calculate the velocity, which was eventually used to determine whether the person was losing mobility. However, this fall detection device also had a high error rate and left considerable scope for improvement.

Pursuing the same line of research, Sixsmith employed a pyroelectric thermal array sensor to detect the movements of people involved in a scene without sensing the background [4]. This could be achieved only when the device was used in the staring mode. The sensor was coupled to a modest processor to obtain a detailed analysis of the person's motion within the detector. The structure of a typical pyroelectric array is shown in (Figure 3.2). This particular type of detector was found to produce a satisfactory and reliable performance.

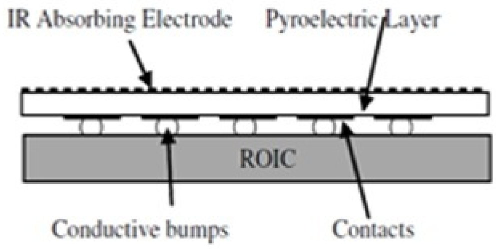


Figure 3.2: Schematic of pyroelectric element [4]

The following sections describe recent developments from portable sensors to video sensors and discuss the issues involved in using video-based fall detection systems. The methodologies used for the implementation of the algorithms of the video-based sensors are also described in detail.

3.3 Types of video-based fall detection systems

Two main types of video surveillance fall detection systems exist: monocular systems and multi-camera systems, which are described in turn.

3.3.1 Monocular systems

A common method for detecting falls is to analyse the person bounding box in the video-taped image [63, 85, 62]. This method can work well when the camera is directed horizontally, but may fail due to obstructing objects. Other researchers [85] have placed the camera high in the room to enable it to cover a larger field of view and to avoid objects that may occlude the view of the camera. Lee and Mihailidis [85] also used the silhouette and the 2D image velocity to detect falls. Furthermore, an elliptical shape representing the person was tracked with a filter to enable the trajectory of the moving object to be used to detect any abnormal inactivity.

3.3.2 Multi-camera systems

A multi-camera system can be useful in reconstructing a three-dimensional representation of the human body shape [86]. The 3D shape was reconstructed from a foreground silhouette in the voxel (volumetric pixel) space. The states of the voxelised person were analysed with a fuzzy hierarchy, i.e. one that can be partially true. For different heights relative to the floor or ground, Auvinet [87] reconstructed the 3D human blob by fusing homographic transformations of the foreground silhouettes in a plane parallel to the ground. The volume distribution along the vertical axis was analysed to identify any events that might seem abnormal, such as a person lying on the ground after a fall. An alarm was subsequently generated when the majority of this volume distribution was amassed near the floor during a period of time.

Thome [88] used a layered hidden Markov model (*LHMM*) to detect falls during walking activities. The *LHMM*, which is a statistical model derived from the hidden Markov model (*HMM*), consists of N levels of *HMMs*, where the *HMMs* on level $i+1$ correspond to probability generators at level i. Every level i of the LHMM therefore consists of K_i *HMMs* working in parallel. The advantage of the LHMM is that it requires a smaller amount of training data to achieve performance than a standard HMM, although some research has shown that *HMMs* can produce better performance with respect to noise sensitivity [89].

3.4 Algorithms used to detect falls

Existing video-based fall detection systems rely partially on visual cues, for instance a sudden change in dimensions [77]. The human shape is very important here, as its deformation can be quantified and used for the automatic detection of falls [90]. The principle is that the human shape can change rapidly during a fall, which is not the case during usual activities, where the deformation is expected to be slow and more progressive. Human shape deformation can therefore be used to distinguish real falls from normal daily activities.

3.4.1 Bounding Box

The aspect ratio of the bounding box [91, 92] is a simple method used to detect a fall. The bounding box is produced by drawing a rectangle around the moving object and also indicates its location on the image; the aspect ratio is the ratio between its dimensions in the x- and y-directions respectively. When the shape of the bounding box changes, these dimensions also change, as therefore does the aspect ratio. This is used to determine whether a fall has occurred [68, 93].

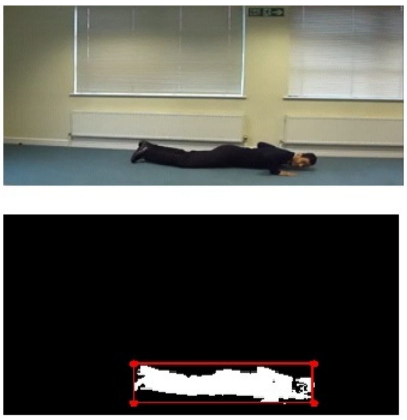


Figure 3.3: An illustration of the bounding box following a fall (original frame is shown in the top figure and the bounding box surrounded by a red line is the bounding box of the person's silhouette

3.4.2 Fall Angle

Previous research suggests that the fall angle comprises a fall detection method [94].The fall angle [88] is defined as the angle between the ground and the person from where it is certain that the person will fall, or the angle between the ground and the person's centroid. An illustration of fall angle is shown in Figure 3.4. When the fall angle is between 45 and 90 degrees, the person is walking, and when it is less than 45 degrees, a fall can be determined [56]. The fall decision therefore depends on the definition of the fall angle and on the style and speed of walking. A disadvantage of using this metric is that if the fall is towards the video camera, it may not be detected.

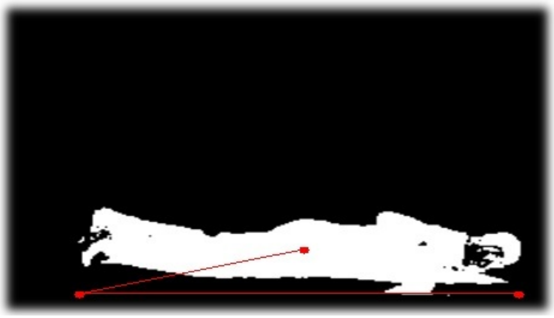


Figure 3.4: The angle subtended by the red lines represents the fall angle of the person (the original image is shown in Figure 3.3; top figure).

3.4.3 Hidden Markov Models

Another class of algorithms, the HMM-based approaches, require some kind of learning phase. The HMM algorithm refers to systems in which the internal state (variable) is unknown and not measurable but has a measurable output [95, 96, 64]. The principle behind the HMM is that a model for each system that can occur is built; from the measured output, the system with the highest probability is the detected system. The hidden Markov model is based on probability theory, according to which, for each frame in the video the horizontal projection histogram is computed from the silhouette matrix to determine the probability that a particular output sequence is produced by a particular system, such as falling and walking. The system with the highest probability value is then chosen. In this method of fall detection, the aim of the HMM training is to generate parameters to characterise each type of system or scenario.

3.4.4 Fall detection using Gaussian Mixture Models (GMMs)

A major problem encountered in fall detection is to distinguish an abnormal event from a dataset of normal activities. This is addressed by modelling the normal activities using a Gaussian mixture model (*GMM*), which represents a weighted sum of Gaussian distributions [97]. The *GMM* parameters are calculated from training data using the iterative expectation-maximisation algorithm or maximum a posteriori (*MAP*) estimation from a well-trained prior model. *GMMs* are normally used as parametric models of the probability distribution of continuous measurements or features in a biometric system.

3.4.5 Other physical features

Horizontal and vertical gradients [87] have also been used to detect falls. Another physical feature is the centroid of the person [98], which changes rapidly when the person is falling.

3.4.6 Extending fall detection methods to include human activity recognition

This section reviews the application of neural networks to fall detection, as this learning algorithm technology has been of great interest to this particular field. Some literature suggests extending fall detection methods by including human activity recognition, such as detecting or identifying a fall. Over the past few years, intelligent systems have played an important role in improving human welfare. Artificial systems perform best while solving complicated computations. Some researchers [99, 86] have employed fuzzy logic to model human activities and to devise sets of rules to define a fall. Other researchers [100] describe a machine-based learning approach to activity recognition in which specific characteristics of the user's behaviour are specified and a learning algorithm is chosen. These characteristics could be the locations of the user's body parts and the angles between adjacent parts, for example. These authors claim that these support vector machines (SVMs) produce reliable results. However, SVMs have the disadvantage that if the number of features is much greater than the number of samples, poor performance is likely to result. Moreover, they do not provide probability estimates; these are determined using five-fold cross-validation, and this performance can suffer. Meanwhile, other researchers [101] have utilised behavioural maps (schematic movements of the human body) to represent the probability of moving in a specific direction and deviation from the learned behaviour to detect abnormal movements or physical behaviours.

An alternative approach is to use a triaxial accelerometer with a threshold algorithm, which raises an alarm when the acceleration threshold value is attained. The use of such sensors can provide an accuracy of 80% and above. [102] constructed a fall detector based on the SVM algorithm. The features used for machine learning were changes in acceleration and the acceleration in each direction. This method allowed falls to be detected with up to 96% accuracy. The researchers also used an accelerometer embedded in a mobile phone, but had some difficulty in fixing this to the body; nevertheless, the system achieved a fall detection rate of 93%. Bourke and Lyons [73] developed a threshold algorithm to distinguish between falls and normal activities such as lying down, sitting down and standing up. In order to distinguish falls, a biaxial gyroscope was fixed to the torso to measure the pitch and the roll angular velocities. This type of fall detection system was found to be 100% accurate in detecting falls.

The next approach to fall recognition is visual detection without posture reconstruction, which relies on the extraction of input data from still images or from video. Any type of computer vision techniques can be applied to the input data, but human posture is not reconstructed. This is in contrast to the visual detection with posture reconstruction approach, which mainly depends on the 3D positions of markers placed at various points on the human body for reconstruction purposes.

Vishwakarma [94] demonstrated a video approach to the detection of falls. After discarding the background of the video, they determined a set of features from the remaining objects' bounding boxes, namely the aspect ratio and the horizontal and vertical gradients. Fall detection was then based on the angle between the bounding box of the object and the ground. This method achieved an accuracy of detecting falls of up to 95% for a single object, but this decreased to 64% for multiple objects. Fu [98] used a temporal contrast vision sensor to extract changing pixels from the background. In this method, the dynamic motion of the object is tracked by a learning algorithm, which reports falls when significant changes occur in the vertical-downward direction.

Sukthankar and Sycara [103] focused their efforts on visual detection with posture reconstruction to build a system to reconstruct the posture of the human body and identify predefined behaviours. The data were obtained by filming 43 body markers with 12 cameras at a sampling rate of 120 Hz. A human body model was constructed from the marker coordinates, then features such as angles, limb lengths and range of motion were computed from the model. The SVM was used to perform learning. This method achieved a fall detection rate of 77% for basic activities such as walking and running. Behaviour was characterised as a sequence of elementary activities and was modelled using HMMs. A number of behaviour models were also defined to enable a new sequence of activities to be classified. However, the accuracy of modelling a particular human body can be compromised by individuals differences in body posture and height, for example, making such systems unreliable.

Huang [5] proposed to detect falls using a video-based algorithm which relies on the back-projected optical flow and on modular neural networks. The proposed algorithm consists of three stages. The first stage is to extract the moving object from the video frame, then to extract the feature point based on the variance of the pixel intensity of the image from this moving object or the human body. The second stage is based on the optical flow back-projection. After extracting the feature point, the algorithm calculates the optical flow based on the Horn-Shunck constraint, which determines a more genuine motion or trajectory. In the final stage, the neural network is used to analyse the motion vector of the featured points to identify the occurrence of a fall. A time-delay neural network using an input feature is employed to train the fall module, as illustrated in (Figure 3.5).

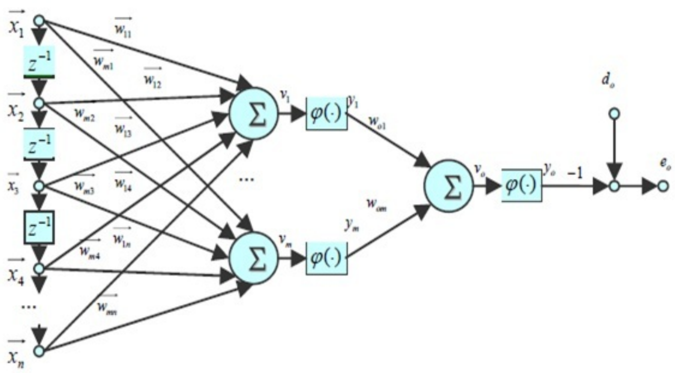


Figure 3.5: Fall module using time-delay neural network [5]

The variables $x(i)$ are input features of the current frame and the previous *n-1* frames where $i = 1, 2, ...n$. The symbol z^{-1}, means delay in time and the notations w, y and e_0 represent the weights of the neural networks, the outputs of the system and the error respectively. However, at the expense of computational time, this may cause a delay in the alarm system for detecting falls.

Related work by Marquis-Faulkes [104] evaluating vision-based fall detection showed that the accuracy of fall detection, the response after a fall has been detected and the privacy of the monitored individual are important factors in implementing a good surveillance system.

Rougier [90] are among those to have used an ellipse rather than a rectangular bounding box to fit the moving body and to have employed ceiling-placed and wide-angle cameras. In order to detect falls, the history of the motion image as well as the aspect ratio of the ellipse and the standard deviation of the angle are employed. The disadvantage of this technique is that sitting down rapidly may be perceived as a fall, while slow falls may be missed. Similarly, Nait-Charif [59] utilised an ellipse-fitting on the foreground area, which was extracted. However, unlike the technique used by Rougier [90], they employed a MAP estimation of Gaussian mixture models.

Juang [105] used both bounding box aspect ratio and projection histograms associated with a neural fuzzy network. A discrete Fourier transform was applied to the horizontal and the vertical projection histograms and a self-constructing neural network system was utilised to classify the postures. The major disadvantages of this method were the length of time needed and the high computational cost. Cucchiara [30] also used projection histograms in their fall detection system, but with the applied classifier built according to an HMM approach. However, shadows, occlusions or carried objects were found to reduce the accuracy of projection histograms.

Rougier [105] employed a different approach, in which the head is tracked in 3D coordinates through the use of a calibrated camera system and a particle filter. For this system, a fall is detected when a large range of motion appears in the horizontal and vertical coordinate systems. Its drawback is that it does not successfully distinguish between a non-fall when the speed of the head is unusually high, such as when the person sits down, and a fall when it is abnormally low, as experienced in slow falls.

3.5 Methods used for the implementation of the intelligent video-based fall detection system in this research

The advantage of a neural network over other methods is that it is adaptable, fast and easy to implement, that it uses a learning algorithm and that the parameters are easily adjusted. The present study incorporated a neural network to produce an intelligent surveillance video system. A neural network of perceptrons was used to compare the silhouette to a set of predefined scenarios or situations, namely standing, sitting, kneeling, lying down, bending or praying. The neural network was trained on these situations or activities, which acted as inputs to the neural network. The back propagation neural network used in this research comprised an input layer, a hidden layer and an output layer representing the distinct situations described above. A supervised learning approach was used to feed the neural network with desired patterns and to allow the system to modify its weights (the parameters of the neural network) according to some learning rules, to identify any type of predefined situation. The mathematical concepts underlying the neural network used in this research are described in Chapter 4.

To maximise the chance of detecting real falls while ignoring non falls, a combination of three important visual detection features the fall angle, the aspect ratio of the bounding box and the hidden Markov model was used on the basis of a voting system. Thus, for each method, an apparent fall was assigned a Boolean value of 1, while 0 was assigned if no fall was detected. If the sum of the values assigned by these three methods was greater than 1, a fall was then deemed to have occurred; in other words, there was a fall in the i^{th} frame only if at least two of the three detection methods returned values of 1 for that frame. This frame by frame combination method, taking account of the positive or negative responses of all three methods, produced a more reliable fall detection system than when using any of the three methods independently.

3.6 Summary

This chapter has reviewed the different types of fall detection systems, ranging from portable sensors to video-based sensors. The underlying technologies and working principles have been explained and compared. The chapter has also elaborated on the various methods to detect falls that are based on visual features, such as the shape or aspect ratio of the bounding box, the fall angle and the velocity or acceleration of the moving body towards the ground. An account was given of the use of hidden Markov models and Gaussian mixture models in detecting falls, along with a detailed review of the current usage of neural network algorithms to distinguish falls from various day-to-day activities. It was found that both visual features and learning algorithms are important if an intelligent and flexible fall detection system is to be produced. The final section explained the methods chosen from those reviewed for the realisation of the proposed fall detection system, to produce a reliable, robust and cheap video surveillance system. Chapter 4 describes the mathematical concepts employed for the implementation of the fall detection system developed in this research.

Chapter 4

Mathematical concepts

Objectives

- To elaborate the concepts used for image processing
- To mathematically define fall detection methods
- To present Hidden Markov Models
- To explain the importance of transition and emission matrices that are found by the HMM train function
- To mathematically describe the neural network algorithms used in this research
- To mathematically evaluate the neural network algorithm using various evaluation metrics

4.1 Introduction

This chapter defines and explains the mathematical concepts used for the implementation of the fall detection algorithm. Firstly, all the image processing techniques that were employed in the processing of the video images are defined, followed by a description of the different types of fall detection methods used to extract certain key features of the silhouette and a thorough description of the Hidden Markov Models based on probability theory. The transition and emission matrices used for the detection phase are also described. A combination of the fall detection methods that are mathematically described in this chapter were used to improve the fall detection algorithm in detecting falls. The algorithms employed in this research for the implementation of neural network algorithms in successfully detecting falls or other activities are also mathematically described. Finally, the evaluation metrics utilised to test the veracity of the neural network algorithm are defined.

4.2 Mathematical concepts used for the image processing

The concepts used to process the video images, such the implemented fall detection algorithm capable of effectively detecting falls, are described fully in this section. The processing involved background subtraction algorithms to detect moving persons or bodies from the recorded scene. This enables the silhouette of the moving body to be extracted from the video being recorded. This extraction was achieved by subtracting the background estimation model from the current input video. By retrieving the person's silhouette from this input video, certain key features of the person can be obtained for the following fall detection stage (Section 4.3). The background image was produced from the background modelling stage, which falls under two main categories, including recursive and non-recursive background modelling methods.

One of the non-recursive methods is the frame differencing technique. In this type of modelling, the frame differencing method takes into consideration the frame at previous time (*t-1*) as the background model. The moving object image is obtained by subtracting it from the previous image shown in Eq. 4.2.1

$$|I_t - I_{t-1}| > T \tag{4.2.1}$$

In Eq. 4.2.1, the variable I_t is the intensity of the image frame at time t, I_{t-1} is the intensity of the image frame at time (*t-1*) and T is a threshold value (the threshold value is an arbitrary value measured in number of pixels to ensure that a satisfied and reliable foreground is obtained from the background).

The next non-recursive method is termed the median filtering technique, in which the background value for each pixel is calculated as the median of that pixel for all frames found in the buffer, which eventually requires a fair amount of memory given by the following formula [29].

$$\text{Memory size} = N \times \text{framesize} \tag{4.2.2}$$

In contrast to those methods, the running average method is a recursive method which employs a fast modelling algorithm and does not require a large amount of memory. The computation of such algorithm is summarised in Eq. 4.2.3.

$$B_{i+1} = \alpha C_i + (1 - \alpha) B_i \tag{4.2.3}$$

In Eq. 4.2.3, the variable B is the background and C_i is the current frame of the video input while the variable represents the learning rate α, which is taken as 0.05. The approximated median filtering technique was used for image processing of the video input, which is described in detail in the following subsection 4.2.1.

4.2.1 Approximated median filtering technique

In this algorithm, when the pixel in the current frame of the video input has a greyscale value greater than that of the pixel in the background, the pixel value in the estimate increases by one. However, when the pixel value in the current frame of the video input has a value lower than the pixel value of the estimated background, the pixel in this background estimate decreases by one.

When the approximated median filtering method is applied to the background model, it converges to an estimate in which 50 percent of input pixels are greater than the background model, while the remaining percentage of the input pixels are less than this background model. The approximate median filtering is described by following a series of equations where I represents the individual frame, while B and F are the background and foreground respectively. Firstly, the algorithm sets the background to the first frame and then resets the foreground to all zeroes.

$$B_{ij}(0) = I_{ij}(0) \tag{4.2.4}$$

$$F_{ij}(0) = 0 \tag{4.2.5}$$

For each time t seconds, D is defined as

$$D_{ij}(t) = I_{ij}(t) - B_{ij}(t-1) \tag{4.2.6}$$

A pixel is regarded as foreground if the difference is more than a specified threshold; otherwise it is considered as background. Initially, the background is taken as the first frame at time $t = 0$, and then it is updated at each frame so that it slowly adapts to the frame.

$$F_{ij}(t) = I_{ij}(t) \quad \text{if} \quad |D_{ij}(t)| \geq \Delta \tag{4.2.7}$$

$$F_{ij}(t) = 0 \quad \text{if} \quad |D_{ij}(t)| < \Delta \tag{4.2.8}$$

(The delta symbol Δ represents the threshold value) The background pixel is increased by one when the current pixel is higher, whereas the background pixel is decreased by one when the current pixel is lower.

$$B_{ij}(t) = B_{ij}(t-1) + 1 \quad \text{if} \quad D_{ij}(t) > 0 \tag{4.2.9}$$

$$B_{ij}(t) = B_{ij}(t-1) - 1 \quad \text{if} \quad D_{ij}(t) < 0 \tag{4.2.10}$$

4.2.2 Mixture of Gaussians technique

In this research, the mixture of Gaussians background modelling was also implemented to examine its performance. In this type of modelling, each pixel of the image is modelled through a number of Gaussian distributions that represent a function distribution F as shown in Eq. 4.3.3. The formula for this algorithm is given below:

$$F(i, \mu, \sigma) = \sum_{i=1}^{n} w_{i,t} \times \eta(\mu, \sigma) \tag{4.2.11}$$

Where η is a Gaussain probability density function. In Eq. 4.2.11, the average of each Gaussian from 1 to k are termed components. This is an estimation of the pixel for the next frame. The weight w and the standard deviation represent the amount of confidence in the estimation. A comparison between the input pixel and the means of the Gaussians tracking that pixel should be performed. The absolute difference between the input pixel and the mean of the Gaussian should be less than the standard deviation of the component which is scaled by a factor D as shown in Eq. 4.2.12. If this is the case, the pixel is regarded as part of the background; if not, it is classified as foreground.

$$|i_t - \mu_{i,t-1}| \leq or \leq D\sigma \tag{4.2.12}$$

After each frame, the component variables w, and needs to be updated [28].

4.3 Silhouette extraction and fall detection methods

4.3.1 Fall angle (Fa)

One important fall detection method is the computation of the fall angle. Based on previous work, the fall angle$\Theta liesbetweenthegroundwhichrepresentsthehorizontalaxisoftheboun$

$$\text{If fall angle is less than } 45^o \text{ then a fall has occurred} \tag{4.3.1}$$

$$\text{If fall angle is between } 45^o \text{ and } 90^o, \text{then normal movement is detected} \tag{4.3.2}$$

4.3.2 Bounding Box

The bounding box, representing a rectangular box drawn around the moving object, can be used to evaluate a certain key feature such as the aspect ratio. The aspect ratio of a bounding box takes into consideration the horizontal and vertical dimensions of this bounding box to evaluate a metric which can be used to determine whether or not a fall has occurred. The bounding box's aspect ratio is the ratio of the length of the box in the horizontal direction to the length of the box in the vertical direction.

$$\text{Aspectratio } = \text{horizontal distance of the BB/ vertical distance of the BB} \tag{4.3.3}$$

When the bounding box substantially changes in x- and y- direction, the aspect ratio also changes, meaning that someone has fallen as it is discussed in details in chapter6.

The variable BB in Eq. 4.3.3 denotes bounding box.

4.3.3 The Hidden Markov Models

The theory of the hidden Markov model is used to model the internal state of a system. The internal state of the HMM can have many states while its exact state is not known to us and also it cannot be measured. However, the outcome or the output of the HMM model can be measured. The hidden Markov model is closely related to probability theory, in which each video frame of the horizontal projection histogram (HPH) was determined. This horizontal projection histogram is computed using the following equation:

$$\mathrm{HPH(y)} = \sum_{\mathrm{x}} \mathrm{S(x,y)}\ For\ all\ x. \tag{4.3.4}$$

In this case, the silhouette matrix is *S(x,y)* and the output is 1 if the pixel belongs to the silhouette or zero if it does not. The Quantised Horizontal Projection Histogram (*QHPH*) sequence is then used to determine the probability that the output sequence was produced by the various types of systems, such as walking and falling. The probability is then compared and only the system with the highest probability chosen. These systems were trained in order to obtain certain parameters that characterise the models of walking, or falling. The training was performed by passing one or more references to the *HMM* train function as well as an initial guess to find the transition and emission matrices. Both the transition and emission matrices were determined by finding the maximum probability of the *HMM* train function in order to generate the reference sequence. These matrices are consequently used for the detection phase.

4.4 Combination methods

Using a voting system, a combined detection method was calculated. This type of combination method states that a fall occurs in the i^{th} frame of the video being recorded if at least two of the three detection method output binary values have a value of 1. The three types of methods were the fall angle, whereby the binary output is called the angle function *faF(i)*, the bounding box output function, which is represented by the *bbF(i)* and the hidden Markov model output which is denoted by the function *hmmF(i)*. These three output functions are then added altogether and compared with the integer number 2. This Boolean result outcome was denoted by the combination of output function *combF(i)*, which means that if *combF(i)* is 1, a fall has definitely occurred, but if *combF(i)* is equal to 0, no fall has occurred.

This was mathematically expressed using the following algorithm.

Begin

Initialise faF(i)

Initialise bbF(i)

Initialise hmmF(i)

Define Sum(i) = faF(i) + bbF(i) + hmmF(i)

Define combF(i)

If Sum(i) is greater than or equal to 2, then assign 1 to combF(i) // fall occurs Else assign 0 to combF(i) // No fall occurs

End

4.5 Neural network algorithms used for identification

This section describes how the neural network algorithm was generated and trained to produce the fall detection algorithm. The principles of the algorithm functions of the implemented neural network are described as follows. The standard architecture of a neural network consists of three layers. Normally, it consists of an input layer, a hidden layer and an output layer. It is therefore only through experiments with the data that one would know how many hidden neurons are needed to achieve an unbiased result. For a basic multilayer neural network, each of the links that connect the inputs have separate weights. The neural network use thresholds (or bias) values or algorithms instead of having to store and update separate thresholds for each neuron (each neuron's activation function is compared to a weighted sum with a threshold as input). These neurons are then connected to the rest of the network and have their own weights which represent the threshold values. This accounts for the weighted sum and the weight of the threshold multiplied by -1. When the weights for the network are updated during back-propagation, the thresholds are automatically updated. The only control over this architecture is the number of hidden neurons as the inputs and desired outputs are already known, so the only decision left is how many hidden neurons are needed in order to determine an optimal amount of hidden neurons to reach the solution(s).

The active function utilised in this neural network is based on the sigmoid function $\varphi(v_i)$ and is represented mathematically in Eq. 4.5.1.

$$\varphi(v_i) = 1/(1 + exp(-avi)) \tag{4.5.1}$$

4.5.1 The Neuron Error Gradients

If the difference between the desired value and the actual value, which is then multiplied by the sigmoid function's gradient, is positive or negative, the desired value increases or decreases the gradient of the activation function respectively. The following equation is used to calculate the error gradient of each output neuron k:

$$d_k = y_k(1 - y_k)(d_k - y_k) \tag{4.5.2}$$

The value at the output neuron k is denoted by yk where yk is the actual output and dk is the desired value at output neuron k. At the hidden layer, the error gradient closely follows the output layer's error gradient. This suggests that for the hidden layer, the error gradient for each hidden neuron represents the product of the gradient of the activation function with the weighted sum of the output layer's errors.

$$d_j = y_i(1 - y_j) \sum w_{jk} d_k \tag{4.5.3}$$

for k = 1, 2, 3,...n.

4.5.2 The Weight Update

The last step in the neural network algorithm is to modify the weights as follows:

$$\mathrm{W_{ij}} = \mathrm{w_{ij}} + \Delta \mathrm{w_{ij}} \text{ and } \mathrm{w_{ij}} = \mathrm{w_{jk}} + \Delta \mathrm{w_{jk}} \tag{4.5.4}$$

$$\text{Where } \Delta w_{ij}(t) = \alpha \times \; inputNeuron_i \times \delta j$$

$$\text{and } \Delta w_{jk}(t) = \alpha \times \; hiddenNeuron_j \times \delta k$$

Where α is a learning rate and δ is the error gradient.This usually represents a value between 0 and 1. The alpha value affects the size of the weight adjustments that consequently affect the learning rate of the neural network. The numerical value has to be properly chosen to enable satisfactory results to be obtained. If alpha value is too low, the system will take a substantial amount of time to learn. If the alpha value is large, the adjustments will also accordingly be large, all of which affects the accuracy of the neural network algorithm as the network will shift constantly from the best solution and will unfortunately be trapped in sub-optimal accuracy.

4.6 Evaluation metrics

In order to evaluate the fall detection algorithm implemented in this research, it was necessary to use some metrics based on probability to test both the reliability and efficacy of this fall detection algorithm. As the output of the fall detection algorithm is binary in nature, the detection of a fall can be positive (denoted by number 1) if the fall detection system identifies a fall or negative (denoted by the number 0) if no fall occurs. Specific tests using statistics (probability) were therefore employed for the evaluation process. Two types of metrics were therefore used to evaluate the fall detection algorithm. The first evaluation metric is termed sensitivity, which refers to the capacity of the implemented fall detection system for detecting and declaring a fall. The second metric is termed the specificity of the fall detection system, which measures the capacity to detect only a fall. These metrics are mathematically defined as follows:

$$Sensitivity = \frac{TP}{TP + FN} \tag{4.6.1}$$

$$Specificity = \frac{TN}{TN + FP} \tag{4.6.2}$$

Where TP represents True Positive, which suggests that a fall actually occurs that is detected by the fall detection system. In the meantime, the notation FN represents False Negative, which means that the fall detection system did not in fact detect the fall. Moreover, the variable TN is short notation for True Negative, where the fall detection system does not detect a fall and no fall has occurred. Finally, FP represents the False Positive, whereby the sensor detects a fall where a fall did not actually occur.

4.7 Summary

In this chapter, the image processing techniques were described mathematically, particularly to explain the background modelling methods. The different types of feature extraction methods were then elaborated and used for the implementation of the fall detection algorithm. Moreover, the principle behind the hidden Markov model was elaborated. This was followed by the algorithm principle of the neural network algorithm, which was used to identify specific activities of the person. The evaluation metrics were mathematically defined and used to evaluate the fall detection algorithm.

Chapter 5

Neural Network

Objectives

- To describe the rationale for the use of neural network in the development of an intelligent surveillance video system
- To present the importance and various applications of neural networks
- To explain how the neural network works
- To describe the methods used for the implementation of a video-based fall detection system

5.1 Introduction

As described in previous chapters, different detection methods were used in this research to detect falls. In this particular one, the neural network algorithm is presented as it was used to develop a robust and reliable fall detection system. Firstly, the rationale for using a neural network in the implementation of an intelligent surveillance video system is presented and the applications of neural networks in daily activities and their significance are described. The working principle of a neural network and the neurone are then thoroughly explained. Following this, an explanation of how the neuron processes an input to produce an output is provided. The activation functions that form part of the foundation of the neural network are described together with the mathematical criteria on which the activation functions are based to ensure that the neurone or neural network output is within certain values. The learning process of a neuron and the learning algorithm behind the working principle of the back propagation method are also defined as they are used for the design of the fall detection algorithm. Furthermore, the procedures used to train the neural network to enable it to adapt following certain input situations are described in order to produce a successful fall detection video surveillance system.

5.2 Rationale for using neural network in this research

The neural network was used to build a more robust video-based fall detection system compared to other existing video-based fall detection systems. The fall detection system should have certain key characteristics, such as the ability to learn from a range of input situations to make the detection of falls more accurate compared to existing methods (such as the fall angle, bounding box, HMM or a combination of them). The incorporation of the neural network algorithm in the fall detection system was aimed at creating an adaptive, reliable and accurate fall detection system applicable to various input situations which include sitting, praying, standing and falling. This fall detection system should also be able to replace the existing commercial expensive fall detection methods to produce quality products.

5.3 The criteria of the neural network for implementing an intelligent fall detection system

For any neural network design, the topology of the neural network is important and depends on an individual's requirements. For instance, the trigger function (such as the sigmoid function) or performance function need to be chosen after careful analysis of the design purpose and an appropriate learning rule should be selected and any conditions should be identified in stopping the training phase. Some commonly used stopping conditions related to neural networks are the desired accuracy, the desired mean square error (namely the difference between the neural network output and one's desired output) and the elapsed time.

5.4 Importance and practical uses of neural networks

Artificial neural networks have recently been shown to be capable of solving complex problems in industry and academia. They are able to solve many engineering problems and intelligent systems and currently play crucial roles in the advancement and innovation of products worldwide [47]. In the medical field, they are used to monitor patients' daily health in hospitals and in all areas of life [92]. In the engineering field, neural networks are used to classify patterns, to develop nonlinear filters adaptable to any situation and are utilised in system identification as well. The neural network is now considered an important artificial intelligence tool [106].

These artificial systems perform much better than linear systems in resolving complicated computations [107]. In addition, when one component or element of the artificial neural system fails to work, the artificial neural system still continues to function without an overall system failure as the neural system has the characteristic property of working in parallel. Normally, the learning process of a neural network relies on the desired outputs and the inputs. Moreover, the neural network does not require any reconstruction as the structure can be used to build on or include new elements or components. The neural network is required to be able to perform simple arithmetic tasks such as additions and multiplications. However, the neural network needs to be trained properly so that it can function well or to the desired response. It is also worth knowing that as the neural network structure increases, a lengthy processing time results.

5.5 Definition and principles of Neural Network

A neural network system was developed based on the working principle of the biological neural networks found in the brain [108]. The construction of an artificial neural network is performed stepwise using a learning rule optimisation process. Firstly, the training input and output data are used to make the neural network system learn and to determine the optimal operating region by using suitable weights at the layers of the neural network system. As the neural network is non linear in nature, this makes the neural network system flexible which suggests that it can adapt itself to different input situations.

Each neuron of a neural network involves a number of inputs, which are translated into its output using a mapping function algorithm. Different topologies can be created by using different types of connections through the neurones of the neural network system. The neuron output, which is determined from its inputs and their temporal behaviours, may differ depending on whether they are working asynchronously, continuously or synchronously. The propagation function is related to a group of variables that define the threshold output bias value and can also be associated with the relative importance of the different inputs and weights.

Neural networks can start to learn by extracting and generalising key features of a presented data training set as these can be correlated to the target response. After the training period is over, the neural net can start to generate the desired output. One advantage of using a neural network is that presenting the neural network system with new input values which do not form part of the data training set can still produce the target output. The learning process of the neural network algorithm is achieved via a training period. This training algorithm can successively change the weights and the bias value of each neurone until the desired result is produced. This desired output is normally defined by the maximum distance between the actual output and the training output. Whilst the back-propagation and perceptron learning algorithm (PLA) can function by restricting themselves to both the local data inputs and outputs of each neurone, global methods such as simulated annealing make use of the overall data such as statistical information.

5.6 Topology of Neural Network

The way in which a topology is formed is mainly based on the connection between the neurons and the number of layers as well as how data is transported throughout the neural network [109]. For example, in a network of processing units relating to a feed-forward neural network system, the data is fed forward from the input units to the output units. Some illustrations of this type of system networks are the Perceptron and the Adaline, in which there are no feedback connections. However, in the recurrent neural network, there are feedback connections where the data is moved forward across the system but an error function is fed back to the neural system to enable it to learn and improve its accuracy in defining a particular situation. From a holistic point of view, the artificial neural networks are regarded as the function networks of a particular topology. A topology is defined as a directed graph that consists of a set of nodes (N). These nodes represent the neurones and a set of transitions (T), which represent directed connections between the nodes in the neural network systems [7].

5.7 A neuron's working principle

A neural network is a structure that can receive inputs and process these to produce the output where the input data can be of any dimensional value. For instance, with a two-dimensional image used in this study, an error was created at the output layer when the input data was presented to the neural network system. This error data represents the difference in value between the real system output and the desired response value. This error was then fed back into the neural network system to adjust its weights through the use of a learning rule. An unbiased and reliable desired result is produced at the output layer when the error is taken back into the neural network system.

In a similar manner to the synapse of a biological neuron, a weight is used to represent the strength of interconnection at the nodes of the neural network algorithm. A negative weight value represents an inhibitory process preventing an increasing behaviour or value while a positive weight designates an excitatory process where the weight favours a positive increase. Figure 5.1 below shows the various components form the actual neuronal activity of the cell. All the inputs'values are added and then modified by the weights. Following that, an activation function regulates the output's amplitude. For instance, this output amplitude may take value 0 and 1, termed binary, or it can be between -1 and 1, also known as bipolar. The variable (vk) represents the neuronal activity and is the sum of all neuronal activities or the sum of the product of the inputs of the neurons (x) with their respective weights (w). Hence, the neuronal output yk represents the output value of the activation function.

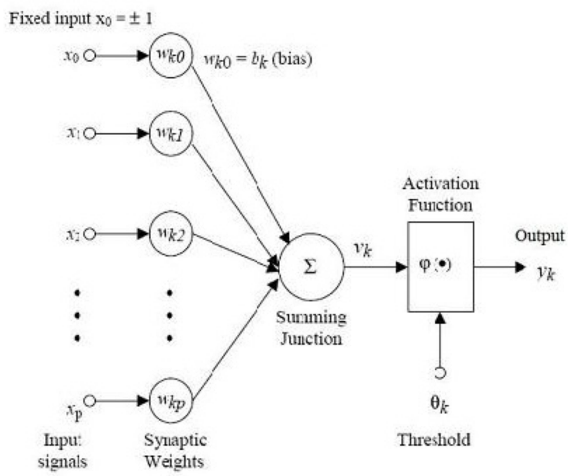


Figure 5.1: Diagram showing the basic workings of a neuron [6]

5.8 The activation functions

Activation functions are denoted using the variable form phi(.). First, a value of 0 is produced by the threshold function when the summation of the inputs is less than a threshold value (v). However, the threshold function produces a value of 1 when the summation of the inputs is larger than or equal to the threshold value. This represents the first type of activation, while the second form of activation function uses a linear function. This function can be binary, taking values of 0 or 1, or in-between these values, which relies on the depending factor. The third type of activation function is a sigmoid function, ranging from 0 to 1, or between the values -1 and 1 [110]. For instance, the hyperbolic tangent function is an example of a sigmoid function.

Derivative of the activation function

The elements of a neural network system (Figure 5.2) are the neuron, weighted inputs (as shown by the arrows pointing towards the cell), and an output (as shown by the outward arrow). The perceptron classifies its inputs into one of the two categories to enable the output of a neuron can take either the value 1 or 0. The schematic diagram (Figure 5.2) displays a basic neuron as a black box as the values of the weights at any particular time are unknown during an operation.

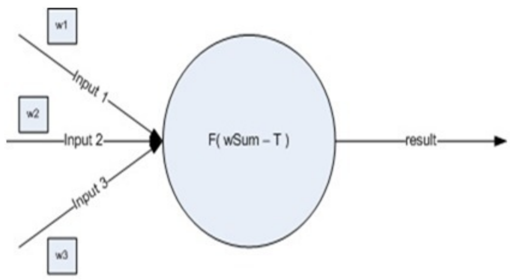


Figure 5.2: Basic neuron as a black box [6]

The neuron consists of an activation function. In this particular case, the activation function is described by using the following notation F(wSum - T) where *wSum* represents the weighted sum of the inputs, and the variable T represents the threshold value. The weights are originally assigned to small random values. During the training period, these weights are then updated so that they are subsequently used to classify pre-defined situations. The weighted sum also depends on the values of the weights and the respective inputs received at the neuron, as depicted in (Figure 5.2).

Different types of functions can be used for the activation function (F). Some examples of the activation function are the step function and the sigmoid function. The step function is a constant function with only finite pieces, whereas the sigmoid function is a function based on the exponential properties [111].

In this research, the sigmoid function was used in the implementation of the back propagation network as this type of function represents the classical activation function. The sigmoid function is a type of exponential function which can either take a value of 1 or 0; this is part of a decision-making process. However, the sigmoid function can never return a 0 or a 1 as it is asymptotic. It is therefore acceptable to take values over 0.9 as 1 and those values that are under 0.1 as 0.

Another important point concerning the input data and the desired target output is that the types of Boolean operations need to be described. For example, the binary OR operator is used to explain the function of the weights and the threshold. With the binary OR operator, the binary output is expected to convey whether an output is true or false to produce a single perceptron with two inputs. In this case, the search space for the neural network is illustrated in (Figure 5.3).

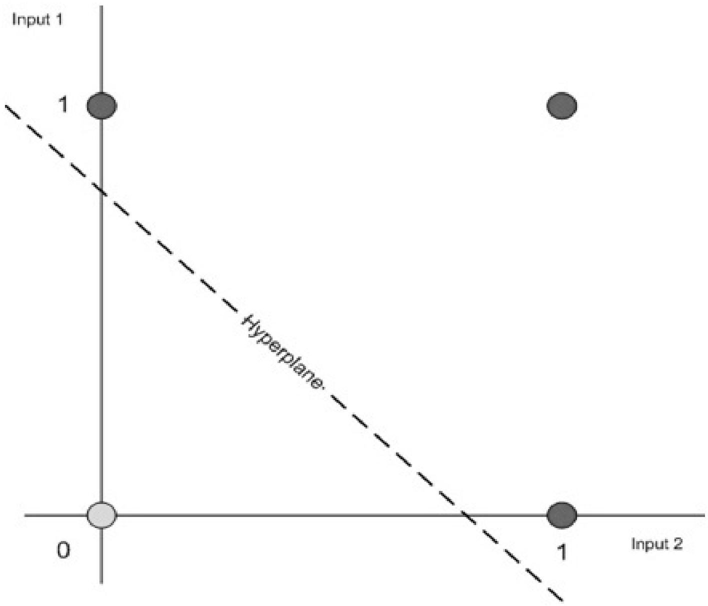


Figure 5.3: The binary OR operator modelled by a single perceptron [6]

As shown in (Figure5.3), the true nature is represented by dark black dots, while the false nature is represented by grey dots. In addition, it is clear that the two classes are separable from this graphical OR operator. A separating dotted line, called the hyperplane, is drawn across the input 1 vertical line and the input 2 horizontal line which separates the two classes. A single neuron can therefore produce a single hyperplane which can be used to solve the above Boolean operation function.

Another important point is that the hyperplane is a straight line, which means that a linear activation function (such as that of a step function) of the neuron is used. However, if a sigmoid function is used, the hyperplane resembles a sigmoid. In simple terms, the hyperplane generated by the image therefore depends on the employed activation function. The threshold value or bias value shifts the hyperplane either left or right, while the weights cause a rotation of the hyperplane. This threshold has to be updated during the stage of the learning process.

5.9 The learning process of a neuron

The activation function firstly processes the input pattern to enable the error between desired and actual values to be computed. The strengths of the interconnections and the weights, are adjusted according to the learning rate and the error. After this process, the neuron starts to learn the next input pattern which is presented to the neural network system. Both the learning rate and the accuracy of the desired output affect neural network performance.

5.10 The learning algorithm of the back propagation method

One neuron is needed to solve linearly separable problems. However, when solving problems that involve at least two classes or where the data cannot be separated, as shown in (Figure 5.4), more than one neuron is needed.

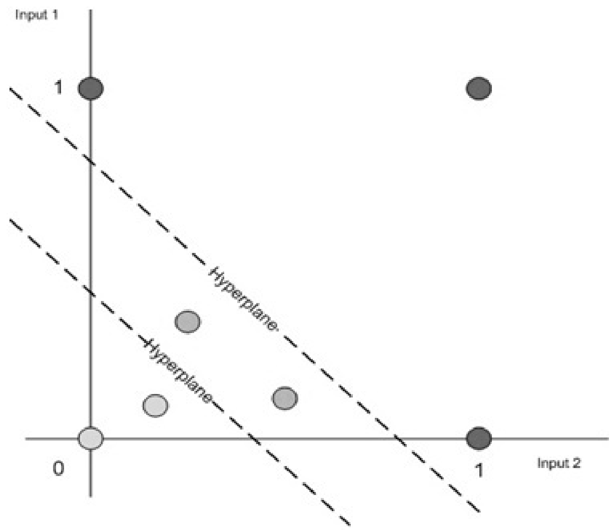


Figure 5.4: Non-linearly separable dataset [6]

As illustrated in (Figure 5.4), two hyper-planes are drawn using dotted diagonal lines to solve a particular problem, which requires two neurons. In order to share inputs and outputs, these two neurons should be connected and in this process a multilayer neural network is created.

The input layer, hidden layer and the output layer together form the basic architecture of a neural network. An important utility of the neural network is that it can solve unknown search space [112, 113]. Ascertaining the number of hidden neurons needed to produce the desired output response can only be achieved through experiments or trials.

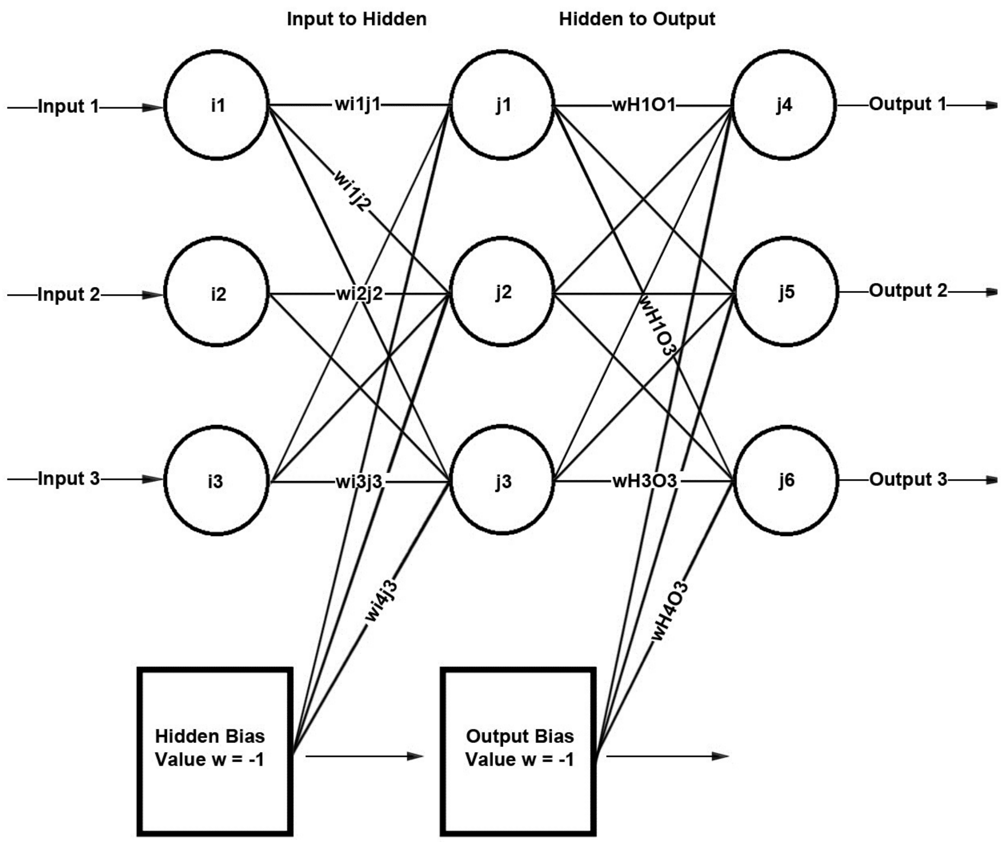


Figure 5.5: a basic multilayer neural network [6]

In (Figure 5.5), the branches consist of separate weights and represent the strength of interconnections between the nodes. The squares, as shown at the bottom of the back propagation neural network diagram, represent two extra neurons with the hidden bias value of -1 and an output bias value of -1 respectively.

The neural network outputs are achieved by adding the weighted sum and threshold weight multiplied by -1. Meanwhile, the weights are adjusted during the back-propagation process and the thresholds are automatically updated [114]. This in turn saves computational time. The number of hidden neurons can be controlled since the data inputs and desired outputs are known. However, a high or low number of hidden neurons cannot generate reliable results. Careful planning and data set testing are therefore vital to find out the ideal number of hidden neurons.

5.10.1 The Neuron Error Gradients

The weights, as part of the neural network algorithm training process, are updated to create the right output in the neural network to reflect a particular predetermined situation. In order to bring the weights up to date, back-propagation is utilised. This implies that once the input is received, the errors are calculated and filtered back though the neural network which subsequently make changes to the weights to reduce any generated error [115]. The weight changes are effectuated using the gradient descent method. The error function follows the high gradient path so that it is ultimately minimised. The error at the output is computed by subtracting the desired value from the actual value, which is then multiplied by the sigmoid function's gradient. The calculation of the error gradient is by Eq. 4.5.2, which is described in Chapter 4 in Section 4.5.1.

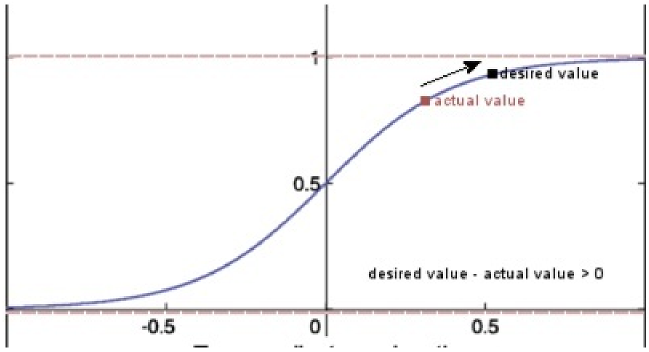


Figure 5.6: the error gradient explanation [7]

The difference in the error gradient at the output and the hidden layers was apparent and is computed by Eq. 4.5.3.

5.10.2 The Learning algorithm

The back propagation learns during a training period termed epoch. The neural network is required to go through several epochs before it has fully learnt all the data presented to it. In this process, there is a need to make sure that the outcome is satisfactory. The steps during a training epoch are described below:

For each input of the training data set, the following is considered:

- data input is fed in (feed forward)
- output is checked against desired value and error is fed back in or (back-propagated) The back-propagation method consists of:
- the calculation of the error gradients
- updating the weights

5.11 The training procedure involved in neural network design

A neural network is designed to enable the application of a set of inputs to produce the desired set of outputs. Several techniques to adjust the strengths of the connections of the neural network system currently exist. Using a priori knowledge is one of these methods to set the weights. A second method is to train the neural system network by feeding it with desired patterns before allowing the neural network algorithm to adapt its weights based on a learning rule algorithm. Two types of learning situations are supervised and unsupervised learning. For supervised learning, the network is trained by providing it with data input and then matching the output patterns [116, 117]. Input-output pairs can be provided to the system either externally or within the system via a self-supervised process.

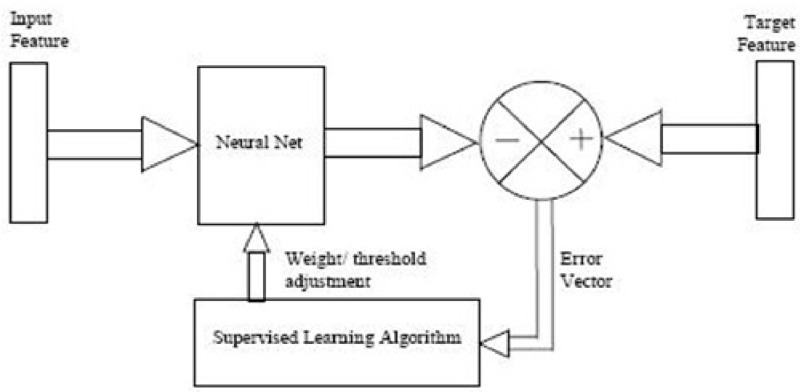


Figure 5.7: A schematic diagram of neural network employing the supervised learning algorithm [7]

The second type of learning situation is unsupervised learning or self-organisation. In unsupervised learning, an output is trained to identify a group of patterns found within the input. Similarly, the system should be able to find striking features or patterns from the data input. In contrast to the supervised learning paradigm, there is no a priori set of classifiers to categorise patterns. However, the system has to generate its own representation of the data input.

The third kind of learning situation is reinforcement learning. This contains a blend of features from the supervised and unsupervised learning methods. The learning machine learns from the exposed environment and also obtains its feedback from the same environment [118]. In order to achieve a balance state, the parameters are adjusted and further changes are thus required within the parameters. Both the supervised and the unsupervised learning paradigms enable the strengths of the neural system connections to be modified according to a learning rule system.

It is important to mention here that almost all learning rule algorithms come from a variation of the Hebbian learning rule system. For instance, the interconnection strength may be more powerful when the two neuronal units A and B are active at the same time. If, for example, unit A obtains data input from unit B, the learning algorithm modifies the weight of the connection between unit A and unit B accordingly with the following change in the interconnection weight δwAB being $\lambda yAyB$, where the variable λ is actually a proportionality constant (positive) representing the learning rate. There is another rule that considers the difference between the actual and desired activation by modifying the weights using the following mathematical change of interconnection strength between unit A and unit B with δwAB being $\lambda yA(dB - yB)$. The notation dB is the desired activation which is provided externally. And this type of rule is also called the delta rule or the Widrow-Hoff rule.

5.11.1 Feed-forward Backpropagation

During the training process, the back propagation neural network will endeavour to correct the mistakes or the errors generated by taking back these errors or mistakes to the system algorithm to enable the neural system to learn what is right or what is wrong. The back propagation method uses the gradient descent technique as a learning method which tries to minimise the error between the current output and the desired response.

5.11.2 Principles of the Genetic Algorithm

Genetic algorithms are another similar method to the working principle of neural networks, and also rely on the type of problem that needs to be solved. However, instead of restricting the parameters of the network, any representation regarded as a fixed length string, also known as a genotype, can be used for a finite alphabet (such as a binary). The interpretation (such as the phenotype) of a string is not important to the algorithm regardless whether it represents the weights and biases of a neural net. The genetic algorithm randomly produces a large amount of those chromosome strings, with each being an individual in the parent population to enable the principles of selection and mutation to be conducted [119, 120, 121].

The fitness of the individuals is effectuated by comparing two individuals for the selection mechanism before making a decision to determine whether tournament or rank selection is preferable. Moreover, the relative fitness of individuals can be determined by providing an error function which categorises each individual according to the average fitness or error of the population. The algorithm will choose individuals with higher fitness or with lower error, which are then propagated into the next generation. The disadvantage of this type of evaluation process is that it is time-consuming, whether the fitness is determined by calculation or experiment. A new generation is created by applying genetic operators to the chosen individuals. The basic operators are mutation, in which one or more digits of the chromosome string are amended, and crossed-over. In the cross-over stage, the two strings are cut and readjusted crosswise to produce two newborn strings [122, 123].

These new strings contain the features of both parents. The reproduction or inversion operators are not particularly important. Until an individual matches the termination criterion, the selection step and propagation step are repeated. Whilst the actual solution is not known, the algorithm converges faster by inputting more data into the fitness function by making the fitness function more linear and accurate. Genetic algorithms are therefore used as a universal tool for most problems requiring optimisation problem that can be represented using a small set of parameters. They are thus often used in modelling, simulations and engineering problems [124, 125].

5.12 Implementation of the neural network algorithm for the development of an intelligent surveillance video-based system

The function *aviread*, an in-built function of the MATLAB programming software, was used to upload recorded videos consisting of the images or frames collected from the falling and non-falling scenarios. These were recorded from the participants in a laboratory-controlled environment. An approximate median filtering tool, described in paragraph 4.2.1, was then employed to compare the substantial changes in the pixel values of the successive greyscale images of the video under investigation. Indeo5 was used to compress the video and the video frame rate was set to 30 frames per second (30fps).

The changes in the pixel values of the consecutive video frames were computed according to a threshold value set at 30 pixels so as to extract successfully the silhouette. The 30 pixels therefore represent the difference of a particular pixel of a frame N to frame $N + 1$ of a particular video where N represents the frame number. The obtained silhouette was hence compressed to a smaller size image picture (for instance from the image size of 300 x 450 pixels to the image size of 20 x 30 pixels) and then quantised for further processing. The aspect ratio of this silhouette, the fall angle and the HMM training of the silhouette were the specific mathematical characteristics used to compare the extracted silhouette with various predetermined scenarios or situations, namely (i) standing, (ii) sitting, (iii) kneeling down, (iv) lying, (v) bending and (vi) praying, using a neural network of perceptrons.

The neural network algorithm was trained based on the presented video frames of sitting actions, praying actions and falling actions of the participants that together represent inputs for the neural network algorithm to start its learning process. In this study, the neural network comprised 600 inputs in the input layer, 10 neurons in the hidden layer, and seven outputs in the output layer that denote the seven distinct situations (as shown in Table 5.1). For instance, the first output to the last output unit of the output layer is denoted by following variables O1, O2, O3, O4, O5, O6 and O7 representing the seven distinguished situations.

If the first output O1 is set to 1 and O2 to O7 are all set to 0, the network will distinguish this pattern as that of a black screen (1000000), and if the first output is '0' and the last is set or activated to '1', then the output sequence 0000001 is considered a praying action. A summary of the binary coding and the desired output response from the neural network system is shown in Table 5.1.

Situation Number	Description	Binary sequence
Situation 1	nothing in foreground	[1 0 0 0 0 0 0]
Situation 2	Standing	[0 1 0 0 0 0 0]
Situation 3	Sitting	[0 0 1 0 0 0 0]
Situation 4	Lying	[0 0 0 1 0 0 0]
Situation 5	Kneeling	[0 0 0 0 1 0 0]
Situation 6	Bending	[0 0 0 0 0 1 0]
Situation 7	Praying	[0 0 0 0 0 0 1]

Table 5.1: The output sequence and its corresponding pre-determined set of situations

To initiate the training procedure, the initial coefficients (weights) for the activation function are arbitrarily taken for the neural network. This algorithm then adapts or updates these coefficients, which improves the neural network system through an iteration process. The whole iteration process took less than 10 iterations to enable satisfactory results to be obtained from the supervised learning neural network.

The following matlab functions used in the construction of the neural network algorithm are described in more detail as follows. The matlab neural network function *newff* creates a feed forward back propagation network. The function has as inputs the input vectors, the target vectors, the number of hidden layers (set to 1 for this particular research), transfer functions for both the hidden layer and the output layer using *tansig* and *purelin* functions respectively. Another matlab function was the performance function, which is based on the mean square error, the backprop network training function *trainlm* and *backprop* weight/bias learning function.

5.13 Summary

In addition to describing the applications of the neural network algorithm, the important criteria needed to design a neural network were stated, together with the type of activation function for a particular situation. The training of a neural network and the type of dataset required to show the functioning of the neural network algorithm for classifying pre-determined situations were also described in this chapter. All the methods involved in the neural network algorithm to meet the criteria were thoroughly explained. A three-layer back-propagation method was used. These layers were the input layer, a hidden layer and the output layer, which may be adapted to represent the seven distinct situations under investigation in this research.

Chapter 6

Preprocessing techniques

Objectives

- To describe the positioning of the camera-based system and requirements
- To describe the conversion of the recorded RGB images to the greyscale images
- To explain the background modelling and subtraction methods used for the processing of the recorded images
- To present the use of the approximate median filtering
- To describe how the silhouette is extracted and how spurious noise in the image is removed through morphological filtering
- To describe the silhouette-based features and motion-based features that were considered to be useful for the detection of falls from the recorded images

6.1 Introduction

Following a description of the working principle of the neural network and the various image processing methods used in this research for the concretisation of the video-based system, this chapter aims to describe in detail the positioning of the camera-based system that was required to obtain optimal quality and relevancy of the recorded images. The results are then described to demonstrate how the coloured or *RGB* images were transformed into greyscale images for subsequent processing. The results emanating from the background modelling and the subtraction methods were then described together with the results obtained while implementing the approximate median filtering. In addition, the production of silhouettes from the processed images are shown and described and the way in which noise was removed from the images is explained. Finally, both the silhouette-based features and motion-based features, which were important characteristics for the detection of falls from the recorded video images, are shown in terms of the results of this research work.

6.2 Positioning of the camera systems and the requirements

The video camera was positioned in such a way that it covered the full range of motion of the participants as well as the full body length of the participant while walking or standing up straight in an indoor environment. These are shown in (Figures 6.1, 6.2, 6.3) for three different situations: falling down, praying and sitting down respectively. The video camera was tilted to the left-hand side to record the motion of the participants. The distance of the video camera and the participant varied slightly in different situations. The distance was about 2.5 metres, as shown in the following figures. Figure 6.1 shows the situation in which a participant fell down on the left-hand side of his body after a lateral rotation of his feet. In order to cover this range of motion, the camera was set at about 2.5 metres from the direction of motion of the participant to enable the latter to easily perform this falling stunt. It was also shown that there were no occluding objects obscuring the camera's ability to capture the range of motion of that particular stunt participant.

MATLAB software was used for the implementation of the methods described in the previous chapters and applied to a video (300x450 pixels, 30fps, compression Indeo5) where a person is moving in a room: walking, sitting down on a sofa, walking again, and then falling on the floor.

Figure 6.1: RGB image showing the time when a participant fell down on the left-hand side of his body after a sudden lateral rotation (Fall5.avi video).

For the praying situation, as shown in (Figure 6.2), where the participant had to perform various movements of bending down, standing up and kneeling down, the video camera was adjusted so that the images from this particular routine or situation could actually depict the various characteristics of the participant's motion and ensure that body lengths were successfully recorded for further analysis.

Figure 6.2: RGB image showing a participant in a praying posture (pray5.avi video)

Figure 6.3 shows the participant performing a sitting down act. The camera was adjusted so that both standing stance and sitting stance were monitored properly for future analysis of video data. The distance between the video camera and the participant was about 2.5 metres.

Figure 6.3: RGB image showing a participant sitting down on a chair posture (sit6.avi video).

6.3 Conversion of the recorded RGB images to greyscale images

In this section, the *RGB* images of the video were changed to greyscale images to enable faster processing of these images. As soon as the recorded video was used as an input, it was subsequently stored in a sequence of frames. Each pixel in the video frame was changed from *RGB* (red, green and blue) colour model to a greyscale (different shades of grey in between black and white model. The *RGB* image was converted into greyscale using a conversion algorithm consisting of the product of weights with the respective colour of red, green and blue, which were added up to produce the greyscale image. This is summarised in the following algorithm (I = 0.299R + 0.587G + 0.114B, where I represents the array of the greyscale pixels of an image, while R is the colour red, B is blue and G is green and their associated weights) [4, 126]. In this research, MATLAB software version 7.0 was used for the implementations of the *RGB* to greyscale algorithm rgb2greyscale... This was consequently applied to a video of 300 x 450 pixels with a frame rate of 30 fps; the compression Indeo5 was used for this purpose. For instance, the greyscale images of a person moving in a room: walking, sitting down on a sofa (see Figure 6.7) and walking again were followed by falling down on the floor.

6.4 Application of the background modelling and subtraction stage

As the background subtraction method involves a wide range of hardware tools, the algorithm utilised to perform this background subtraction method is approximate median filtering. This subtraction method normally involves medium complexity compared to other background subtraction methods. The frame difference is of lowest complexity and the Mixture of Gaussians method is of greatest complexity. The frame difference is of lowest complexity as it consists of fewer computations as compared to the other methods. For the approximate median filtering, each frame is compared with the background image and the greyscale value difference of the pixel is then computed. Moreover, if the difference of the pixels is more than a particular threshold value (such as 30 pixels), it is considered as a foreground; if not, the difference in the pixel value is considered as a background. For instance, the corresponding greyscale conversion of the images from (Figures 6.1, 6.2, 6.3) are shown below to depict the various shades of grey of the video recorded images. As observed, the content of the images was not altered and they were useful enough to extract the silhouette or profile of the moving body. These greyscale images also showed that in this particular algorithm colours were not useful data but were still important for the creation of the greyscale images.

Figure 6.4: Greyscale image showing a participant falling down on the left-hand side of his body after a sudden lateral rotation (Fall5.avi video).

Figure 6.5: Greyscale image showing a participant in a praying posture (pray5.avi video).

Figure 6.6: Greyscale image showing a participant sitting down on a chair posture (sit6.avi video).

6.5 Application of the approximate median filtering stage

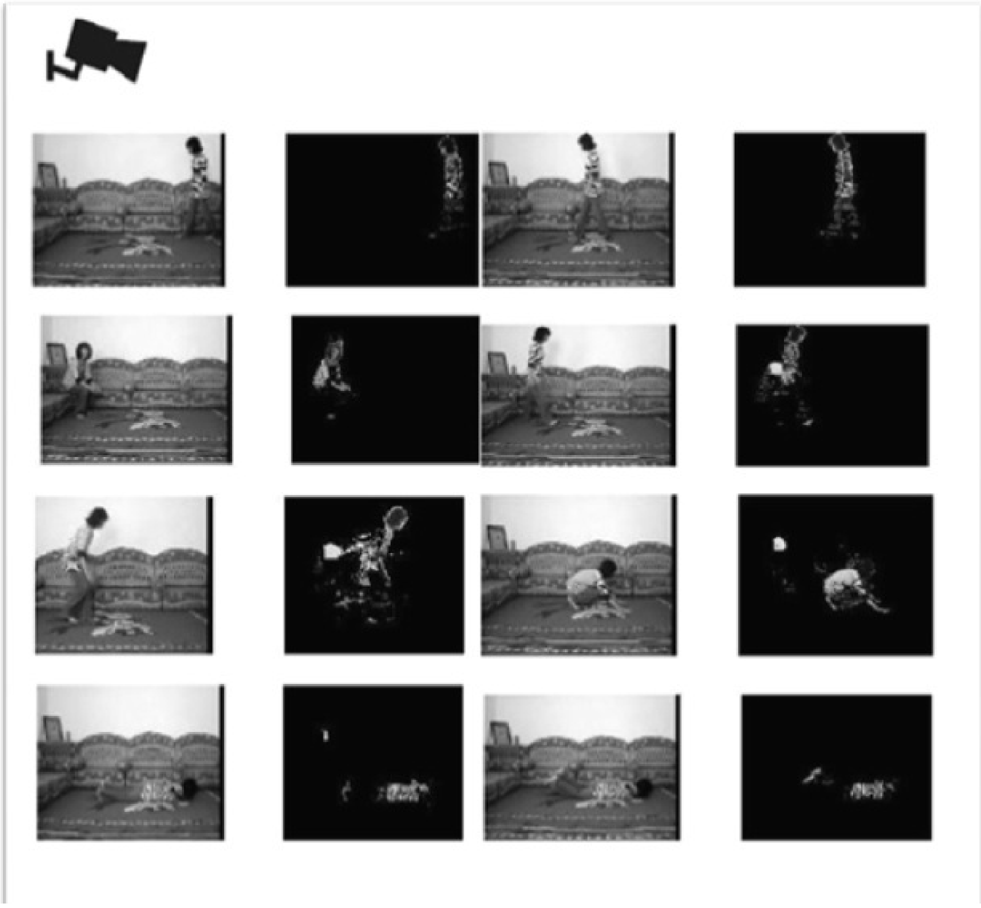


Figure 6.7: An array of images and the corresponding silhouette

Starting from the top of (Figure 6.7), the first row represents the images of a another person (different motions and different persons) walking (greyscale). Next to this image is the corresponding silhouette, which shows the profile of the person walking. The second row of (Figure 6.7) shows a person sitting down on sofa and walk again together with the corresponding silhouettes. The third row shows the person starting to trip over or falling down, together with the corresponding silhouettes. Finally, the fourth row shows the person lying down together with the associated silhouette.

These images represent screenshots taken at discrete time intervals. The left side (greyscale image) shows the original frame and the extracted foreground (silhouette) is on the right. The approximate median filtering provided promising results in terms of achieving a compromise between speed and accuracy. The computation time for the video was about six seconds, far less than the video time, suggesting that it would be possible to use this algorithm for real time fall detection. The approximate median was compared with the mixture of Gaussians. The mixture of Gaussians adds complexity and time but does not show better results. Some time was also spent tuning the threshold Delta, which was finally set to 30. This threshold determines whether a pixel is foreground or background. A value too high for Delta results in fewer false positives (namely less noise in the extracted foreground) but also more false negatives (some parts of the subject are mistakenly assigned to the background. A value too low for Delta results instead in more false positives (more noise) but fewer false negatives (more or less all that is expected in the foreground is actually marked as foreground). The methods dilation and erosion or a combination of both were used to clean the silhouette.

6.6 Extraction of the silhouette

The disadvantages of the wearable devices lead to the use of video sensors. However, video sensors also have their difficulties. Due to privacy being the main concern when using video sensors, the person should be aware of the fact that he or she is being filmed. The elderly should be assured that it is impossible for other people to view the video films, which are processed solely by computer. In addition, to safeguard privacy, the raw data are not stored and only the silhouettes are used to track the elderly [63]. After having isolated the foreground, the silhouette was then extracted. This is a simple operation which requires the production of frames with white pixels for the foreground image and black pixels for the background. The silhouette extraction is conceptually a further stage in video processing, but in the MATLAB code it was implemented at the same time as the foreground separation. In the MATLAB code, the silhouette is stored in memory not as an actual video but as an array of frames, each of which is a matrix in which the elements can be either 0 or 255 (namely black or white). This is to facilitate the implementation of the detection algorithms that can already use the extracted information as a matrix, rather than having to process a video file again.The following figures show examples of extracting the silhouette for limping and stumbling scenarios.

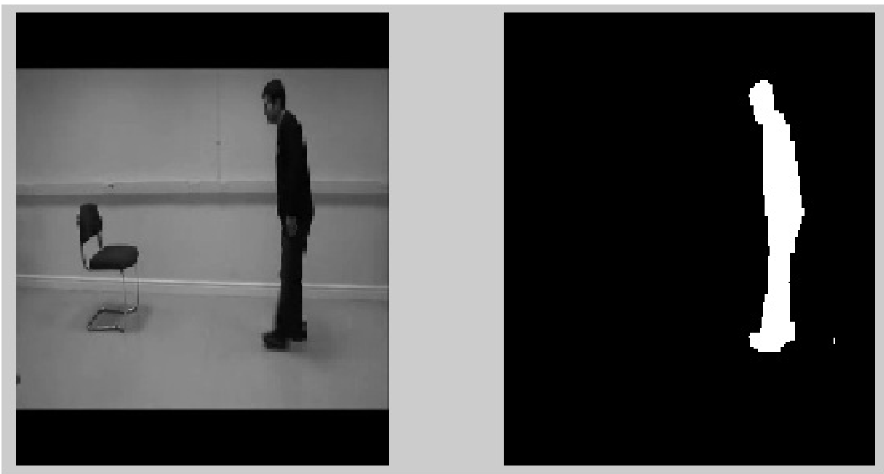


Figure 6.8: Example of extracting silhouette for limping participant

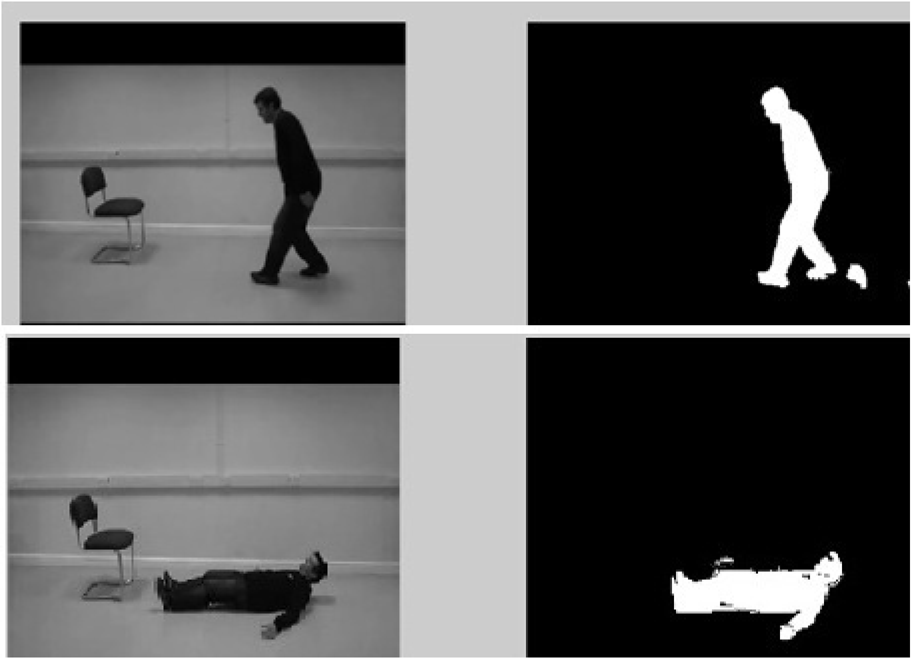


Figure 6.9: Example of extracting silhouette for stumbingscenario

6.7 Characteristics of the silhouette-based features and motion-based features

In this section, the characteristics of both the silhouette features and the motion features are described and contrasted. These features are important as they allow the fall detection algorithm to learn the characteristics of fall, to distinguish between a fall and a non-fall and to ensure that a fall has definitely occurred through a confirmation of 'supposed fall' features. The silhouette based features are described in the following sub-section.

6.7.1 The silhouette based features

The main purpose of extracting features is to give a thorough description of the position of the moving object. The features help to distinguish between different activities. This study aimed to distinguish fall activity from other non-falling activities, such as walking, sitting down or lying down. There are two main groups for all the features. The first consists of silhouette-based features and the other of motion-based features.

Silhouette-based features can be extracted from still images and are outlined as follows: (i) The aspect ratio: The ratio between the height and width of the bounding box. (ii) The fall angle: Defined as an angle between the ground and the person from where it is certain that the person will fall. (iii) Height of the bounding box.

6.7.2 Motion-based features

There are a set of features extracted from the object and its bounding box, such as the aspect ratio, falling angle, horizontal *(Gx)*, and vertical *(Gy)* used in the fall model.

- Aspect ratio: The most simple and effective feature to vary a normal from an abnormal pose is the aspect ratio of the bounding box. This rectangular box can be drawn around the moving object such as a person. The dimension between x and y directions is termed the ratio of the bounding box. When the bounding box substantially changes in x- and y- direction, the aspect ratio also changes, meaning that someone has fallen [68, 67].

- Fall Angle is one of the simplest methods used in the literature and refers to the angle between the ground (horizontal axis of the bounding box) and the person's centroid. A fall can be estimated when the angle is less than 45 degrees [56]. In addition, it is assumed that is between 45 and 90 degrees when the person is walking depending on the style and speed of walking. In this implementation, it was assumed that the person is falling when value is less than 45 degrees. The fall decision is therefore based on the definition of fall angle. However, this method can be unreliable in the case of falling towards the camera unless new methods based on silhouette or motion-based features can be devised to dissociate between falling towards the camera and walking towards the camera.

6.8 Summary

This chapter described the camera positions according to systems and requirements. In addition, the techniques underlying the conversion of an RGB image to a greyscale image were described together with illustrations using the stunt participants performing the falling actions. The background modelling and the subtraction methods utilised for the processing of the images collected from the video-based fall detection system were described. A thorough explanation of how the silhouette is extracted was provided. Finally, the silhouette-based features and motion based features were explained as they were important techniques for fall detection from the recorded images from the fall detector.

Chapter 7

First approach to fall detection

Objectives

- Presenting the Importance of silhouette extraction and feature extraction
- Implementing of fall angle and related results
- Implementing of bounding box and related results
- Implementing of hidden Markov Models and related results
- Discussing the purpose of HMM training
- Combining of fall angle, bounding box, HMMs and results

7.1 Introduction

The previous chapter described the results obtained from the image processing on the video images of fall scenarios. In this chapter, the various fall detection algorithm techniques are described together with the related results. Elderly people can easily forget to wear common portable fall detection sensors, or sometimes feel irritated by having to take them wherever they go. This prompted the investigation of the promising video-based fall detection method, which preserves privacy and gives them freedom in their daily physical activities. However, in order for video surveillance systems to be efficient and accurate, not only do they need to be robust with regard to image processing difficulties, but features should also be extracted for analysis purposes or to detect falls.

7.2 Fall angle results

The approach proposed in this study was implemented by using MATLAB as the programming language. Video was captured in a room in which the camera was placed on a table with a horizontal viewing angle. The video clips were obtained during all possible types of fall (Sideway, Forward, Backward) indoors and with occlusions avoided to obtain a clear view. All the clips included one moving object (person). Generally speaking, the approach achieved accurate results. However, false alarms can be given, for example, when the person prays or does some exercises. In addition, falling towards the camera cannot be detected. The following result was obtained where column2 shows the result of the fall angle method:

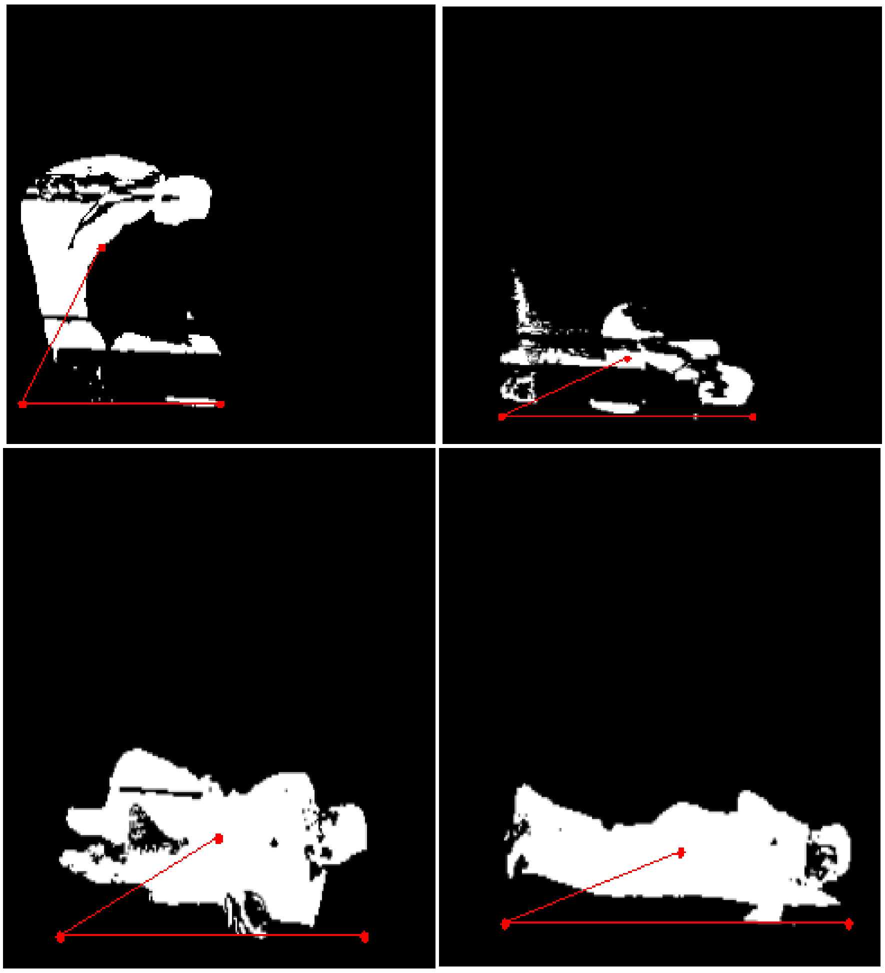


Figure 7.1: An example of fall angle detection.

Figure 7.1 illustrates how the computed fall angle evolved throughout the duration of the reference frame. The x-axis represents the frame number and the y-axis represents fall angle (radiance). The subject enters the scene, and before they appear in the film (from frame zero to about number 140) the silhouette is empty and the fall angle is 1 (a nominal number value with no physical meaning). Moreover, when the angle is between the centre of the mask of the body and the ground is less than 45 degrees, the fall is detected. The fall occurs between around frame 450 and 600 frames.

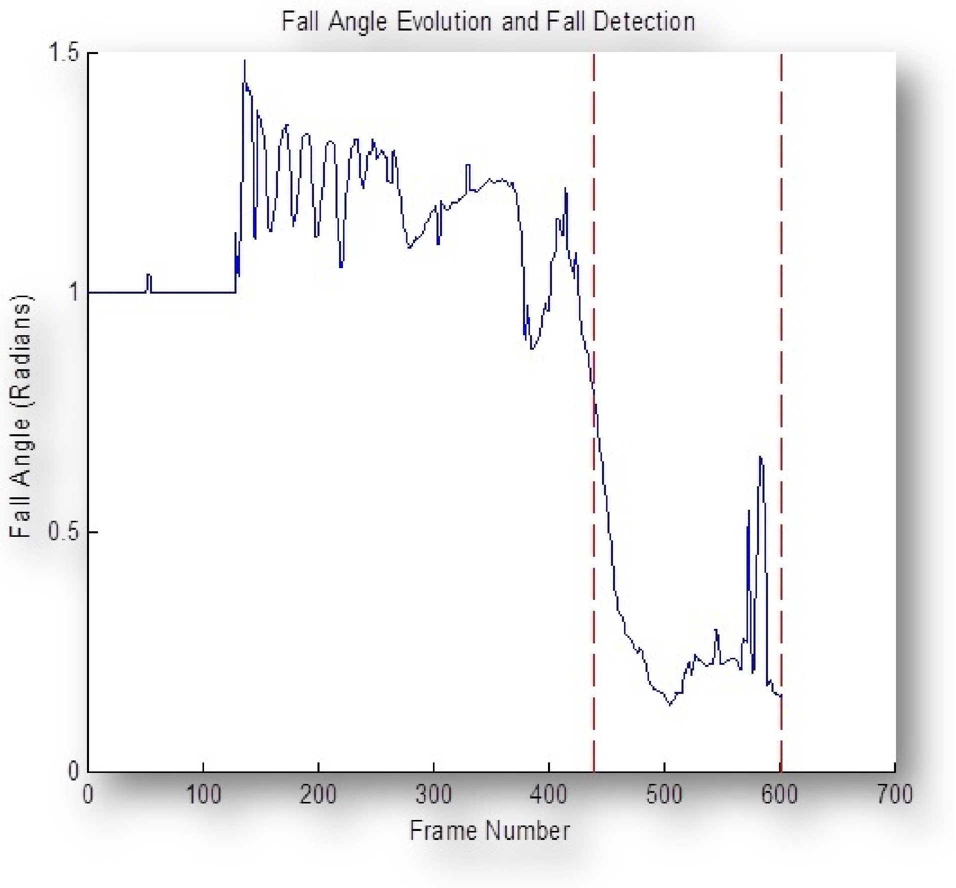


Figure 7.2: Fall angle evolution and fall detection.

7.3 Bounding box results

After acquiring the silhouette, the next step involved feature extraction. Initially it was found that the width to higher ratio of the silhouette of the bounding box provided adequate fall feature detection. If the silhouette is larger in the vertical plane versus the horizontal plane, the bounding box width to height ratio is indicated. This is typically the case when the person is standing or walking as both cases are the same. In this result, the bounding box is drawn around the person in red.

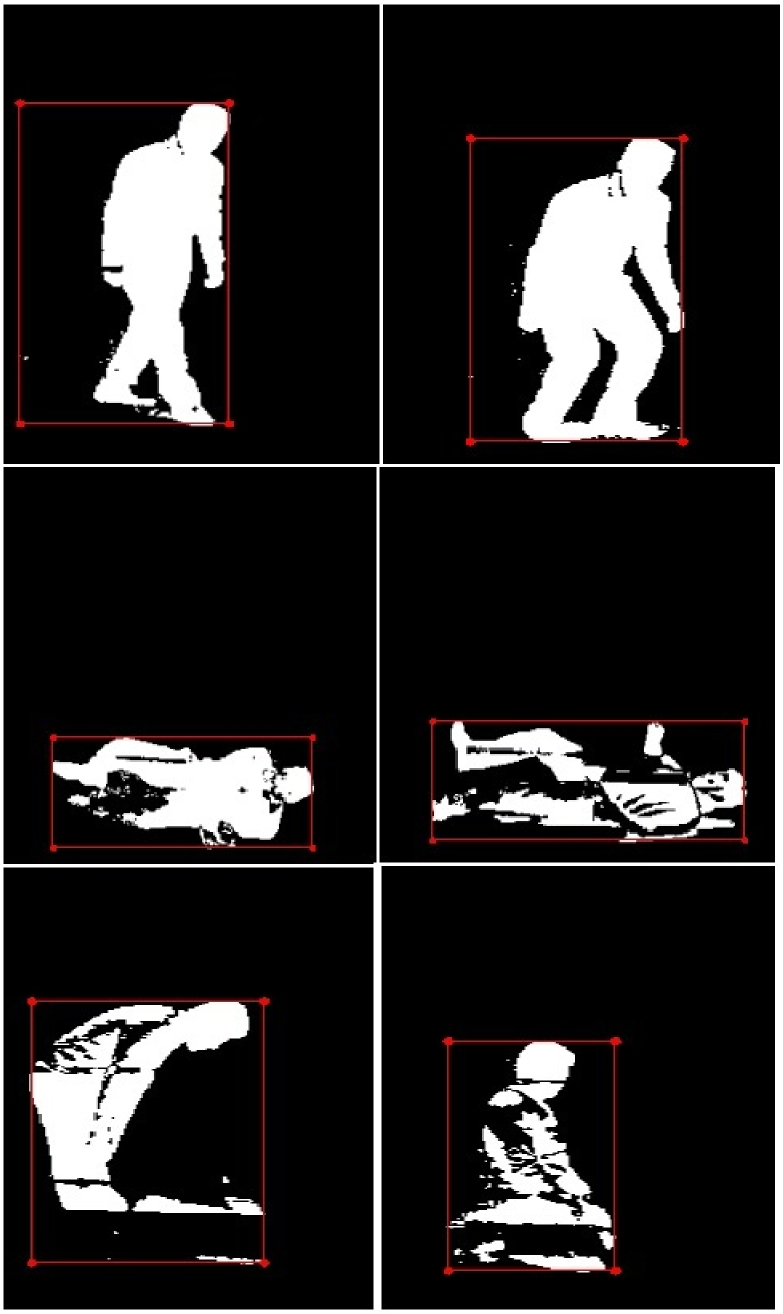


Figure 7.3: examples of the bounding box.

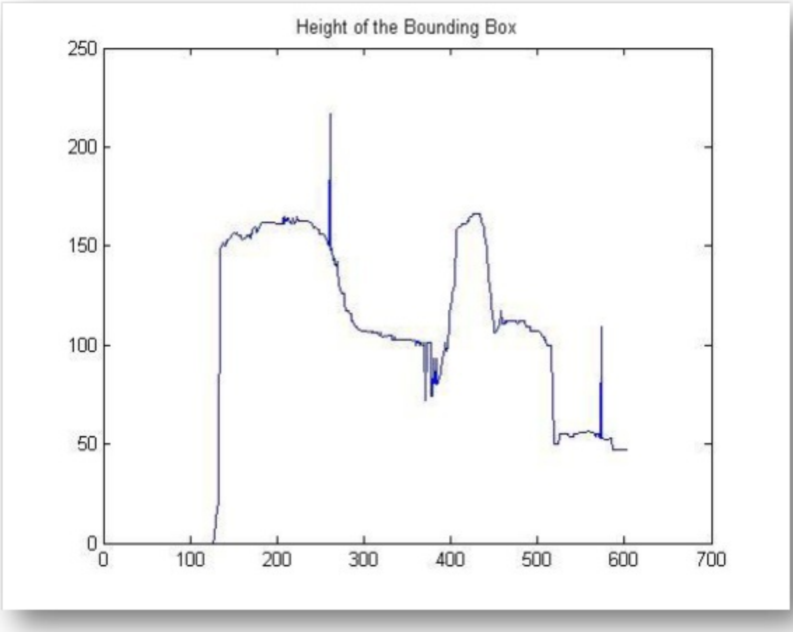


Figure 7.4: Height of the bounding box

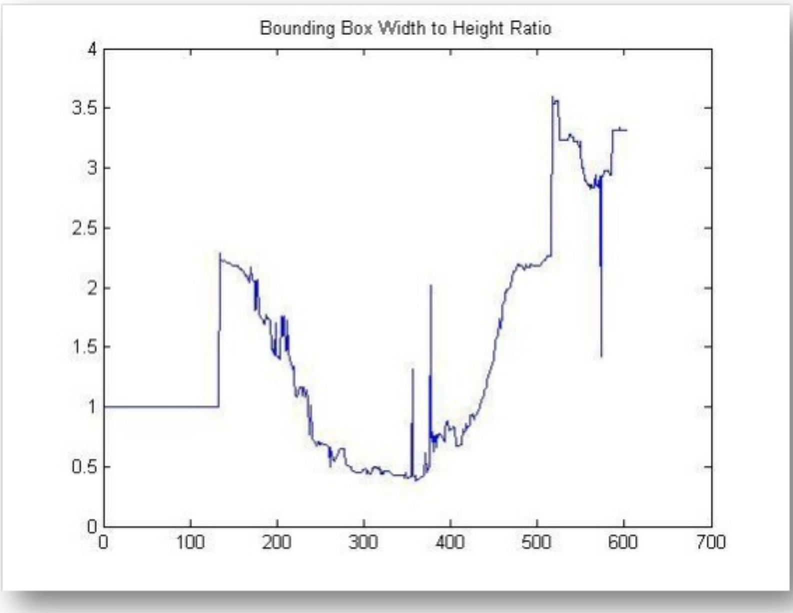


Figure 7.5: Bounding box width to height ratio

In the following, the x-axis represents the frame number whereas the y-axis represents the bounding box features. The lowest point of the bounding box, the foreground, is defined when the person stands on the floor and the height and the width can be calculated by capturing the heights when the person walks. The person's height is calculated over time and recorded. The height is short when the person bends or sits on the sofa.

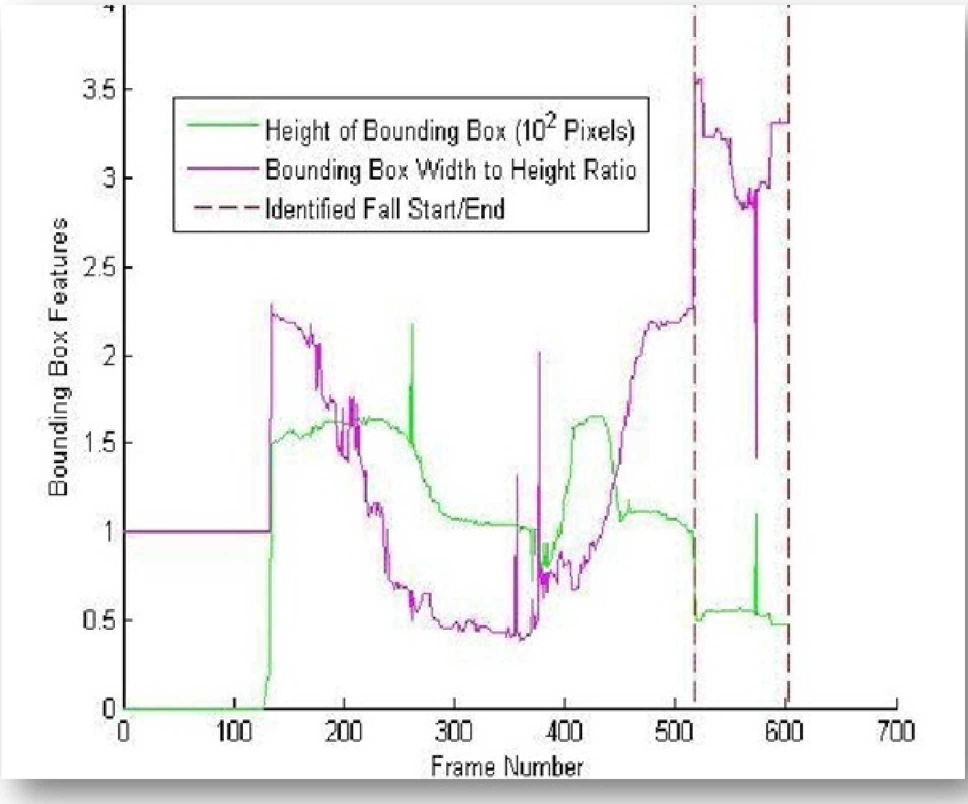


Figure 7.6: Bounding box result

In the first, no any calculation is noticed in the figure as the first frame belongs the background only and the subject has not yet appeared. When the subject starts to appear in the scene, the width to height ratio becomes larger. In (Figure 7.6), the height is shown in green colours and the bounding box width to height ratio is shown in red. In the frame number around 110, when the subject started walking into the scene, the width to height ratio is small. It may be noted that the height changes from frame to frame because the person is walking, sitting, kneeling, and then falling on the ground. The width to height ratio in between frames around 510 and 600 suddenly changes and becomes much larger. This happens when the person is on the ground and the ratio is greater than 2.5. The frame is represented in red dashed lines in the frame.

7.4 Results of Hidden Markov Models

The theory of the hidden Markov models underpins the algorithm implemented in this study. The theory is related to systems for which the internal state (varying among a finite number of states) is not known and not measurable (hence the term hidden) while the output is measurable.

This theory is successfully used in other types of heuristic algorithms, for example recognition of speech [127]. The overall idea is to construct one model for each "system" or case that can occur (for example a spoken syllable), before taking the measured output (for example a fragment of speech) and process it for understanding, for which system the probability of having generated that output is higher. The system with the higher probability is hence the detected system.

The theory of hidden Markov models follows from the theory of probability and is quite general. For example, it holds true regardless of how a system is defined or how many states a system has. As the theory leaves quite a lot of flexibility an improper choice of the parameters may lead to a poor detection algorithm.

In this case, "walking" and "falling" were chosen as the systems, the output unit passed to the algorithm for detection was the quantized horizontal projection histogram (QHPH) calculated on the video frame, and there were two internal states for each of the systems.

For each frame in the video the horizontal projection histogram *(HPH)* was calculated as described in Eq. 4.3.4. The sequence was then quantised with these intervals:

$$\begin{cases} QHPH(y) = 1 & \text{if } HPH(y) = 0 \\ QHPH(y) = 2 & \text{if } 1 \leq HPH(y) \leq 5 \\ QHPH(y) = 3 & \text{if } 6 \leq HPH(y) \leq 10 \\ QHPH(y) = 4 & \text{if } 11 \leq HPH(y) \leq 20 \\ QHPH(y) = 5 & \text{if } HPH(y) \geq 21 \end{cases}$$

The QHPH sequence was then given to the function that calculates the probability that this output sequence was generated by each of the two systems, ”walking” and ”falling”. The probability was then compared and the system with the highest probability selected. It is worth noting that separately the two systems had to be trained to get their parameters.

The algorithm is fast, able to detect the fall, but does not perform as well as the other two algorithms. Detection happens at a late stage in the video when the person has already been on the ground for some time.

It should be noted that the two systems had to be separately trained to obtain their parameters. The following figures show some snapshots of the HMM result:

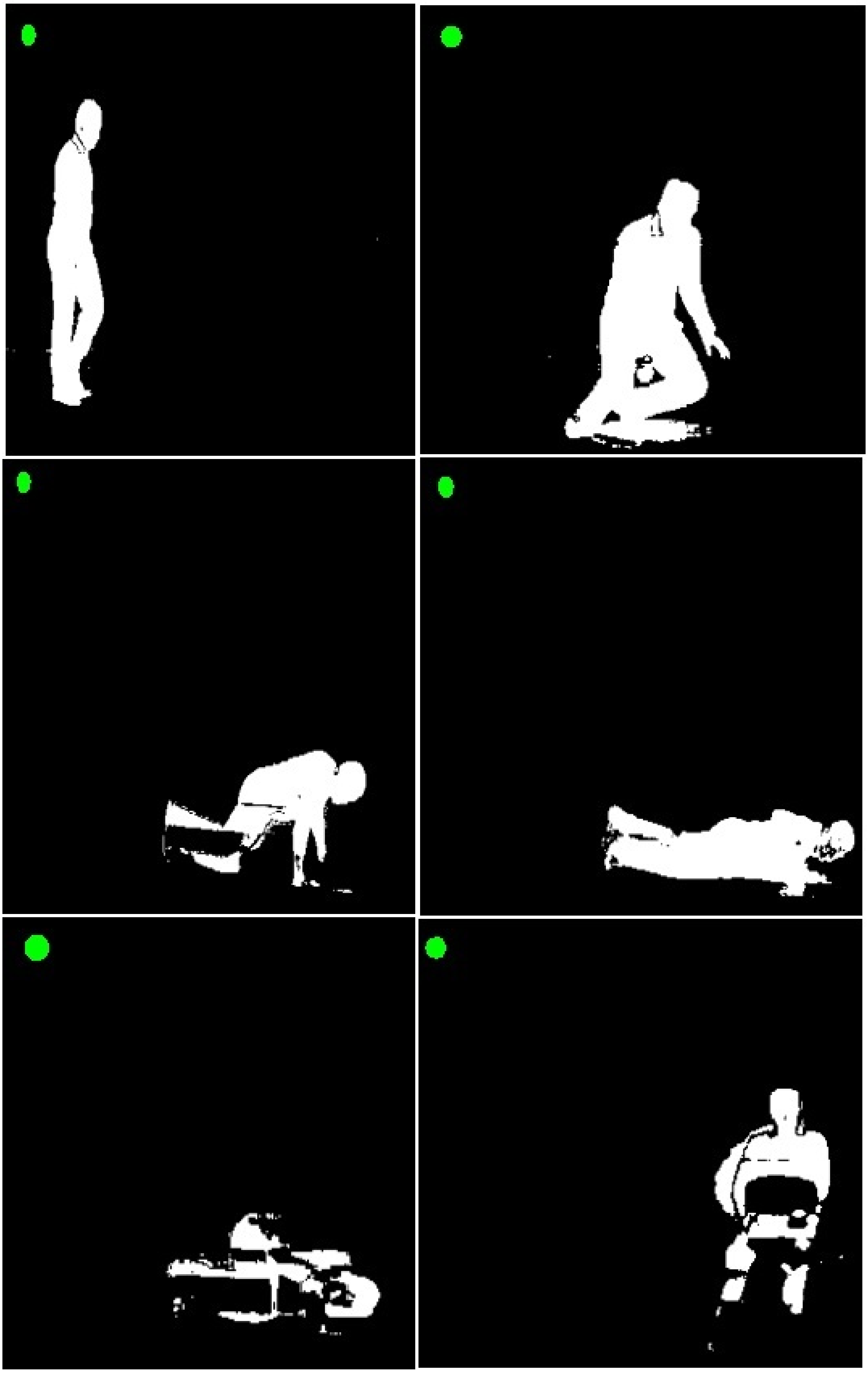


Figure 7.7: A snapshot of some HMM examples

7.5 HMM training

The purpose of the HMM training is to produce the parameters characterising the two models for walking and for the falling scenarios. The training happens by passing one or more references to the hmm train function, together with an initial guess of the transition and the emission matrices. The initial guess values for 'walk' and 'fall' were used respectively as a snapshot from the following code:

trans_guess_walk=[0.95 0.05; 0.10 0.90];

emit_guess_walk=[0.99 0.01 0 0 0 0; 0.1 0.2 0.2 0.2 0.2 0.1];

trans_guess_fall=[0.95 0.05; 0.10 0.90];

emit_guess_fall=[0.99 0.01 0 0 0 0; 0.05 0.05 0.1 0.3 0.3 0.2];

The *HMM* train function finds the transition and emission matrices with the maximum probability of obtaining the reference sequence. The matrices obtained in this way may then be used in the hmm function for the detection phase. One transition and one emission matrices for each of the two models (walking and falling) was obtained.

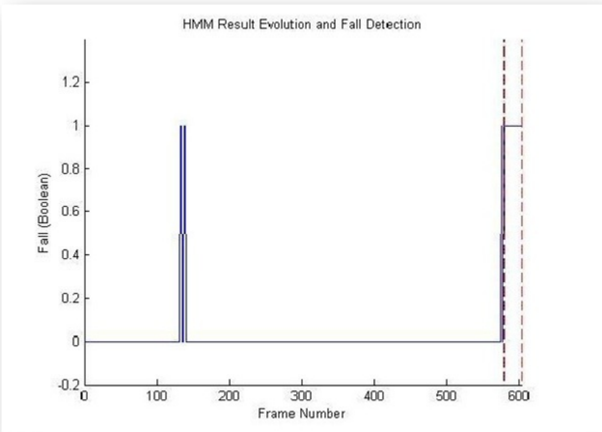


Figure 7.8: HMM evolution and fall detection

7.6 Combined fall angle, bounding box, and HMMs

Finally, the combined detection was computed by a voting system in which it is stated that there is a fall in the i^{th} frame if at least two of the three detection values have a value of 1. The three values *faF(i)*, *bbF(i)* and *hmmF(i)* were added, this number compared with two and the Boolean result assigned to *combF(i)*, so that if the sum is greater or equal to 2 *combF(i)* will be equal to 1. This means that a fall occurred; otherwise *combF(i)* equals 0, meaning that a walking scenario was detected. In the following, the silhouette is printed, with a coloured point added in the top left corner: green if *combF(i)* is 0 (walking) and red if *combF(i)* is 1 (fall). The current frame and fall angle data are plotted. This result gives a more reliable system rather than using one of the other methods separately.

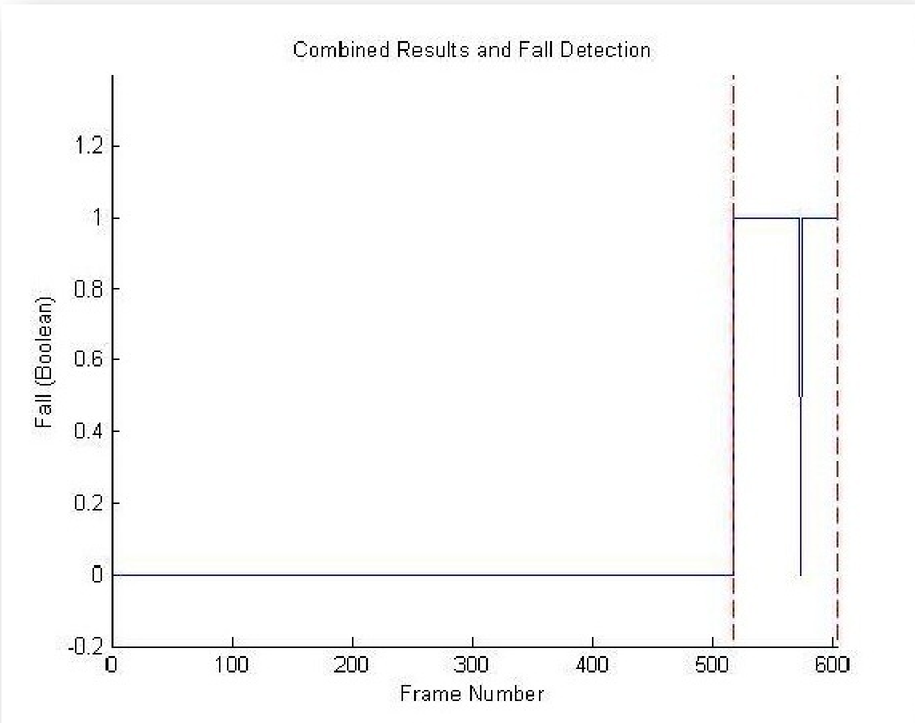


Figure 7.9: Combined results and fall detection

Results from the last system are very good for the reference film used in this study. However, before using this algorithm in production systems, effort should be made to understand the strengths and weaknesses of these methods.

The strengths are:

- Fast (simple computation) - apt to be used in real time detection scenarios.
- Reliable in the detection of a silhouette going down on floor.
- In the fall angle method, the critical angle needs to be able to be adjusted to allow better detection of certain situations instead of others.

However, this system has disadvantages, such as:

- There is no state information: only one frame can be considered at a time and there is no accounting of what happens in more than one frame. For example, there is no accounting of the speed at which the fall happens, which instead could give important information to avoid false positives. Another example is that after a detected fall there is no mechanism to process the later frames to confirm that the person cannot stand up again.
- In the fall angle, to allow detection the person has to fall in the direction perpendicular to the camera. If the fall is toward the camera, there may be no detection. The detection could potentially lead to different results for different positions of the camera.
- No accounting for the location of the person; there is no possibility to specify special areas where a drop in the fall angle should not be detected as a fall, for example a bed or a sofa.

7.7 Summary

In this chapter, the first approach methods that were implemented and used for the detection of falls were described together with the associated results. The methods used were based on the silhouette features and consisted of fall angle, bounding box method, Hidden Markov Model and finally a combination of them. Using a combination of these methods makes the detection of a fall more certain as compared to only one method. The following chapter describes a second approach using a neural network algorithm. It extends the current theory and algorithm of the existing fall detection system by identifying all the situations and by distinguishing between a fall from various other situations such as sitting, praying and standing.

Chapter 8

Discussion of the second approach (neural network) results

Objectives

- To identify every motion of the subject for each frame
- To discuss the issues after extracting the silhouette
- To show how the neural network is able to learn new situations
- To show how the neural network interprets the activity for each frame
- To demonstrate the ease of implementing our algorithm codes

8.1 Introduction

Due to a large number of accidental falls occurring to elderly people annually, which can unfortunately lead to many health complications, and even to death, it has become a priority to investigate the production of appropriate fall detection systems, which should consist of a number of key properties. These fall detection systems should be cheap to implement, require minimal maintenance, be highly efficient and reliable, and also preserve the privacy of their users. The solution proposed here is based on a video fall detection system whereby the video sensors monitor only the person's silhouette, which is in fact a binary map (black and white pixel image) of the person's movements. In this video-based detection system, the actual *RGB* image of the person is never stored. An approach is utilised to merge the situation identification together, frame-wise, with the semantics of the situation during the evolution of time. The extracted silhouette, with an original size of about 300 x 450, is scaled down to a size of 20 x 30, then planar-quantised so that a situation sequence is generated and subsequently compared with a certain number of predefined situations.

The neural network algorithm is used for the silhouette recognition; this activity recognition is state-based and interprets the activity for each frame, together with information about the time at which each event is happening. Therefore, without question, the proposed technique implemented in this research delivers a much better and more reliable fall identification activity than other techniques used by other researchers. Anderson and Lin were unable to differentiate fall activities from activities comprising walking, falling and kneeling, and could not validate their methods in terms of distinguishing a lying down situation from a falling situation [63]. In fact, their fall detection system achieved only a 52.5% recognition rate, as they used only one key feature α for fall recognition.

Research conducted by Jared revealed that the application of these limited fall detection methods in real-life situations are limited because there are a great range of fall actions, as well as a large number of false negative falls which the system should not categorise as falls [77]. Anderson also stated that while such a model-based fall detection system was able to detect different actions using different hidden Markov models *(HMMs)*, it was not able to deal with unknown situations. In addition, *HMMs* have some drawbacks, such as involving many more computations based on statistical techniques [63]. In contrast, the neural network, a recently developed technology, is based on a heuristic approach, in which the results are better than those of *HMMs* because it works with input vectors which are trained to determine the weights or coefficients of the neural network. Experiments in this research show that a multilayer perceptron *(MLP)* neural network is appropriate for the classification and the recognition of human biomechanical motions. The next section will first discuss the extraction, and then move on to explore the importance and the function of situation identification using neural networks.

8.2 Key Features of the Fall Detection System

An approach is used to combine situation recognition with activity recognition, frame-wise, using the semantics of situations over time evolution. The silhouette is first scaled and planar-quantised, this then generates a situation sequence, which is compared with a number of known situations whereby the activity in question is identified by the neural network. The video fall detection system involves the following stages: background modelling and subtraction, silhouette extraction criteria, fall detection metrics, training of the neural network, neural network detection and activity detection.

8.3 Situation Identification using Neural Networks

In order to identify as accurately as possible a human fall, one should be able to identify every motion of the subject for each frame. This identification of movement should be merged also with the evolution of events in time to give an enriched and reliable information of what is happening to a person (whether the person is falling, sitting, kneeling, bending, praying, lying down). The situation refers to what is happening to the individual at every instant in time and the activity refers to the identification of the sequence of activities by the trained neural network algorithm. For instance, if a person is standing and suddenly kneels down to collect something and then stands up again, this scene may be represented frame wise by the following sequence: Frame 1 standing, Frame 2 standing, Frame 3 kneeling, Frame 4 standing , Frame 5 standing and this sequence can be classified as a no fall activity. However, if another series of sequence consists of the following: Frame 1 standing, Frame 2 standing, Frame 3 Kneeling, Frame 4 lying down, Frame 5 lying down, Frame 6 lying down, this in turn can be associated with a positive fall activity and will in fact starts an alarm. After the silhouette has been extracted and normalised, a neural network of perceptrons was used to classify it against the set of predefined situations.

8.3.1 Silhouette Normalisation

While extracting the video, two problems arose and these were that the silhouette can be in fact anywhere in the picture frame and secondly the size of the silhouette is unknown. Therefore, these two issues should be solved in order to use the silhouette effectively as an input in the situation identification algorithm. The bounding box was used to surround the silhouette from the picture frame, then it was positioned to the middle of a new image of same aspect ratio of the original frame whereby the height and the width of the bounding box becomes the height and the width of the new image. The working of finding the size of the image is summarised as such. Let h and w be the height and width of the bounding box and let Fh and Fw be the height and width of the whole picture frame. If the ratio of the height to the width (h/w) is less than (Fh/Fw) then it means that the bounding box's width will become the width of the new image but the height of the new image will be $w \times (Fh/Fw)$. If the ratio (h/w) is greater than (Fh/Fw), then it means that the bounding box's height will become the height of the new image and the width of the new image will be $h \times (Fw/Fh)$. Then, the image is resized to the 20 x 30 pixels after the selecting the bounding box and fitted into the image of the same aspect ratio of the initial frame.

8.3.2 Normalisation Results

The following two images show the results of the normalisation of one sample frame. The left image depicts the extracted silhouette surrounded by the bounding box (in red). In the right image, the content of the bounding box is centred and resized to fit into a smaller image of a chosen normal size (20 x 30 pixels). The value of the greyscale of the pixels in the smaller image is a vector of 600 elements (20 x 30), which will be used as input for the neural network.

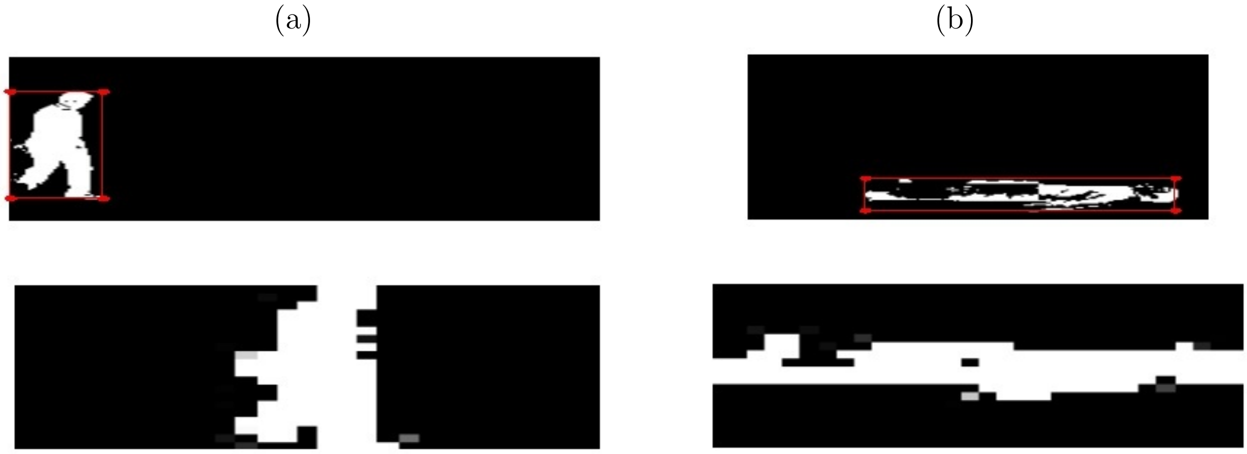


Figure 8.1: Examples of Silhouette Normalisation

Another example of silhouette normalisation is shown in figure 8.1-b, where a person is walking; again, the region of interest is bounded by a red box and then centred, as shown in the bottom figure.

8.3.3 Creating the Neural Network

A neural network is a network of simple elements (neurons) that are able to provide simple identification/classification of input. In order to allow the correct operation of the network, a set of input vectors with the corresponding (desired) output must be provided to the training algorithms, enabling the best parameters for the network to be determined. For this step, MATLAB's Neural Networks Toolbox, which provides a set of functions to work with neural networks *(NN)*, has been used.

This line creates the network:

net = newp(NN_P,NN_T)...;

NN_P is a matrix containing sample input vectors. *NN_T* is a matrix containing sample corresponding output vectors.

As mentioned previously, the first step in developing the neural network is creating the neural network. In this case, there are seven situations at a given instant of time: null, standing, sitting, bending, kneeling, lying and praying. A lot of the data was captured in the lab, and comprised several movies, each of which contained a number of different situations. There was a vector of 600 elements, as mentioned before, which will constitute the input for the neural network. The input data signal was acquired as a two-dimensional array (20 x 30), while the input to the neural networks was a one-dimensional array. The neural network will find relationships of its own. In addition, there is one hidden layer of 10 neurons and output as a vector. The outputs are the seven situations.

8.3.4 Training the Neural Network

The neural networks have two main functions: pattern classifiers and nonlinear adaptive filters; in this research, they are used to categorise certain predefined scenarios. The artificial neural network is an adaptive system; each parameter is changed during its operation, and is deployed to solve the problem during what is normally called the training phase. The artificial neural network is developed using a systematic stepwise procedure that establishes reliable optimisation, using a criterion known as a learning rule. The input/output training data is crucial for the neural networks, as it reveals information which is essential to finding the optimal operating point. In addition, owing to its nonlinear nature, the neural network processing elements make it a flexible system.

This neural network system receives the input information, processes the data and produces the output. This line launches the network training:

[net,tr] = train(net,NN_TRAIN_P,NN_TRAIN_T); NN_TRAIN_P and NN_TRAIN_T...

represent the matrices of the chosen input and the corresponding output vectors to be used for the training. The training of the network is a procedure by which the best-performing internal coefficients are determined for the neural network. To the training procedure, a number of input vectors is provided and the corresponding desired-output vectors (in machine learning, this type of training is called supervised learning).

The training algorithm starts by taking random coefficients for the network and then changes each coefficient so that the performance of the network is gradually improved. This is an iterative procedure and each iteration makes the coefficients slightly more efficient. The iteration loop is usually carried out until no further improvement is being made. In typical neural networks, this may be anywhere from ten to ten-thousand iterations, but a few hundred is common. In our implementation it took less than 10 iterations.

In our case, input consists of video data array. As soon as the corresponding input is presented to the neural network, the target response is set at the output, an error is produced between the target response and the output. This error information is fed back to the neural system that adjusts the weights systematically and this process is iterated until the desires output response is produced. For the neural network training stage, this involves first of all to set the seven predefined situations to certain values and these were as follows: the SIT_NULL = 1, SIT_STAND=2, SIT_LIE=3, SIT_SIT=4,SIT_KNEEL=5, SIT_BEND=6, SIT_PRAY=7. The function eye (7) was used to generate the sequences and identify each preset conditions. Couple with that the following movie or frames were chosen to train the network and these are as follows:

selected_frames='pray5' [ ] [ ] [ ] [ ] [ ] [ ] [ ]

’pray6’ [ ] [ ] [ ] [ ] [ ] [ ] [ ]

’pray7’ [ ] [460 491 514] [ ] [ ] [ ] [429] [308]

’fall5’ [ ] [36 43] [91 101 111 128] [65] [ ] [ ] [ ]

’fall6’ [6 27] [86] [139 145] [ ] [ ] [ ] [ ]

’fall7’ [13] [101] [212] [ ] [131] [ ] [ ]

’sit5’ [ ] [114] [ ] [202 215] [ ] [ ] [ ]

’sit6’ [ ] [66] [ ] [294 298] [ ] [114 127] [ ]

’sit7’ [ ] [74 175] [ ] [ ] [ ] [ ] [ ] ;

The first one was empty because the first situation was the background and this is why is null. The neural network perceives each layer as an array. The first array represents the inputs and are multiplied by the weights which are unknown coefficients to get the desired output and this output is compared with the threshold. This threshold is determined by the number defined in the activation function.

Among the selected frames or videos, the following clips pray5, pray6 and pray7 are videos representing scenarios of praying; also fall5, fall6 and fall7 are scenarios of falling and finally sit5, sit6 and sit7 are sitting scenarios conducted by various healthy volunteers or participants. Now the *silhouette.mat* file is fed into the program and then a bounding box enveloped the silhouette and it is normalised as well as scaled to the following dimensions of 20 x 30 (height x width). Then the aspect ratio of this bounded box is computed in order to find out if the binary silhouette corresponds to a fall and this is used to train the neural network for all existing or predefined situations.

8.4 Neural Network Detection

The following figure shows the MATLAB window that appears during the training of the neural network. This screen appears automatically when you launch the MATLAB train function. In the top part it shows a block diagram of the network (Layer) with the vector of the weights (w), the vector of the biases (offsets) (b) and the step function that determines if the output will be 0 or 1. W and b together constitute the parameters that the training will try to optimize. The Algorithms section shows the algorithms used for the training and the computation of the performance. The Progress section shows the number of iterations (Epochs), the elapsed time and the value of the performance. The buttons *Performance* and *Training State* allow plots of the evolution of the network during the training. When the neural network is produced and the coefficients have been optimised through training, the neural network can identify the person's activity at any instant in time and the following code **NN_Y= sim (net, NN_P_ ALL)** is used to identify what is happening over the whole scene video. The output vectors that were chosen were as in table 5.1. Finally, the neural network is used to identify the situations and the situation vector is saved in an output mat file. The final output formula used to obtain the variable y is given by: Y = [1 2 3 4 5 6 7]*NN_Y

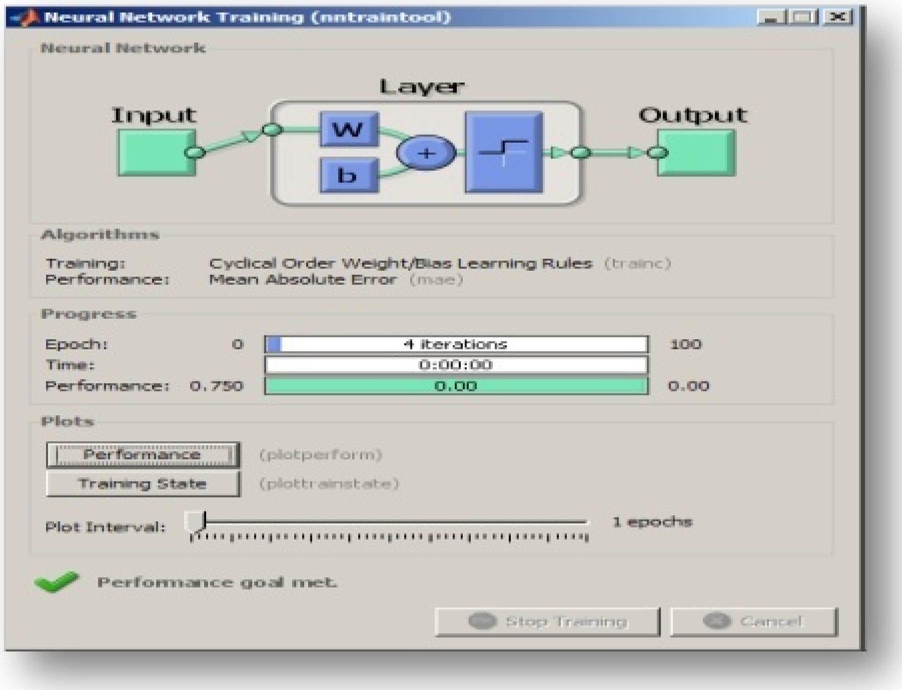


Figure 8.2: The neural network training to find and optimise the weights for prediction of the fall scenarios.

The neural network is a very popular pattern recognition algorithm and is used in many scientific fields for predictions. In this research, the neural network was able to identify the situations within a few iterations and with only one hidden layer and simple architecture.

The activity detection is state-based and interprets the information emanating from the various frames together with the information about the times of the various situations that occur in time. The neural network is used for the activity recognition as it is a good technique in delivering a better and reliable identification for fall situations. The alarm count is set to 90, and the some predefined constants are STATE_NULL is set to 0, STATE_LYING is set to 1 and STATE_ALL_OTHERS is set to 2 and the matfile containing the situation vectors are loaded into variable y and the alarm as well as the counter is set to 0. Then each situational vector is associated to the corresponding STATE constants. Moreover, an alarm starts when the counter is greater than the ALARM_COUNT when the state is equal to the conditions STATE_LYING and STATE_NULL but the counter is set to null when the state is equal to both STATE_LYING and STATE_ALL_OTHERS. If the alarm is greater than 0, an alarm is produced. In this activation detection stage, an alarm seems the appropriate option to attract people attention and also to summon help faster.

8.5 Identification of the Situations

After the frames, which represent the situations or events, were chosen, the neural network was used to identify what was actually happening during this video session. The identification of the situations (in this case pray7) was first found in the following manner. As usual, the silhouette is bounded by the red box and then scaled as shown in the second top right graph. The first blank (black) box is a NULL situation; hence, there was no bounding box for this situation.

8.5.1 The Identification of Praying

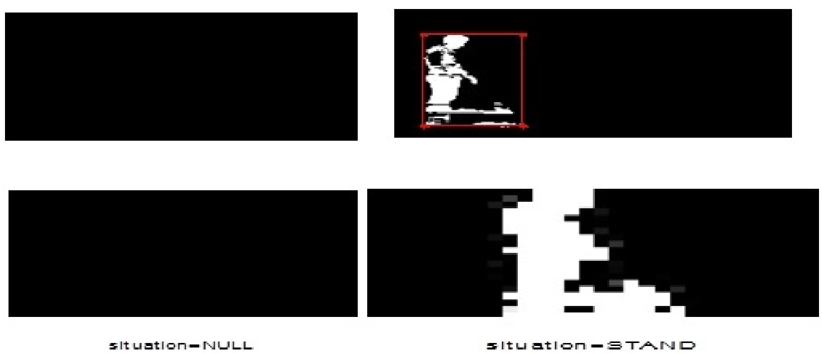


Figure 8.3: Null and standing situations with their normalised silhouette.

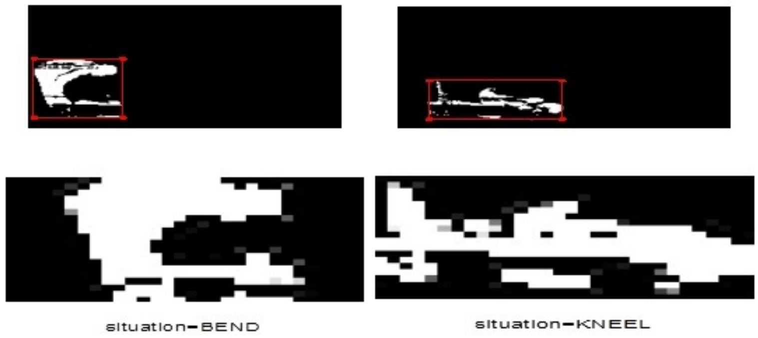


Figure 8.4: Bending and kneeling situations with their normalised silhouette.

In (Figure 8.3), the left figures represent a NO situation and its corresponding normalised NO situation, and the right ones represent a person standing with the bounding box around the silhouette of the person, and (at the bottom) the normalised silhouette of the person.

Figure 8.4 depicts bend and kneel situations with silhouettes and their corresponding normalised figures. In the top left figure, the person is bending, and the silhouette (top) and its corresponding normalised figure (bottom) are shown. The right figures represent the kneel situation of a person (silhouette and normalised).

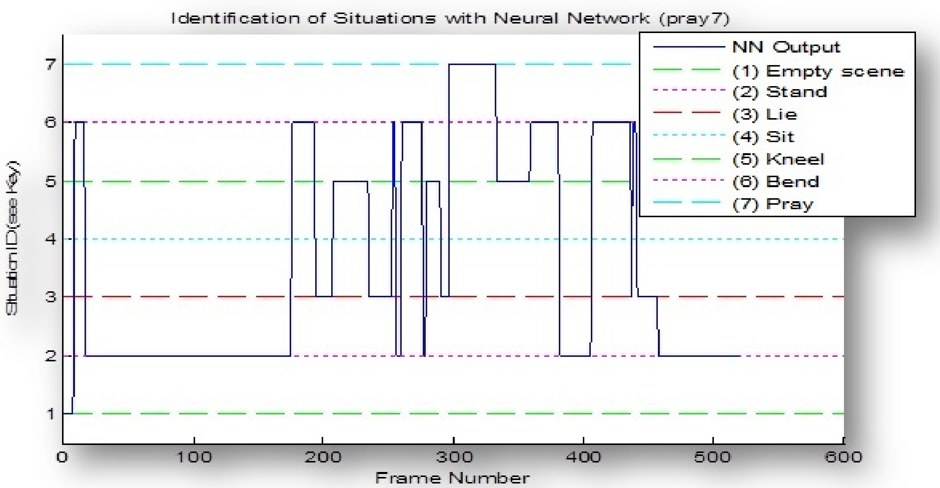
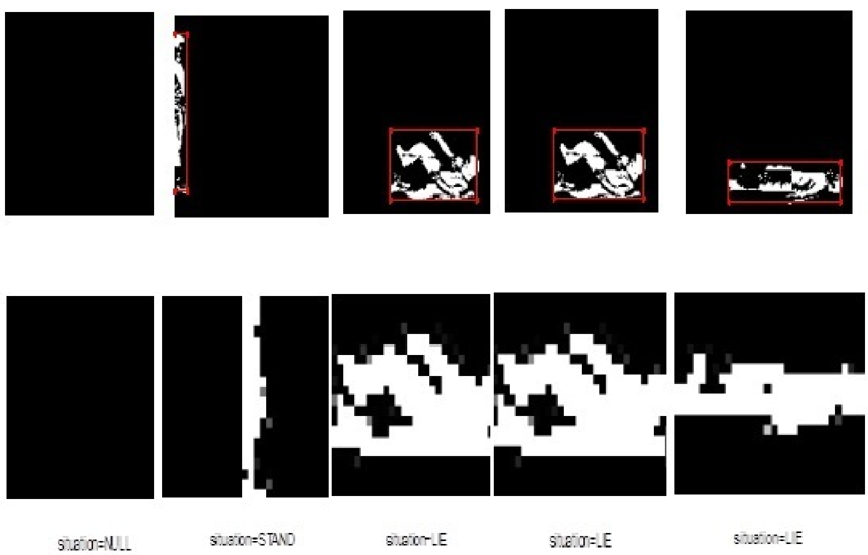


Figure 8.5: PRAY situation identification.

This graph represents a movie containing 600 frames, and reveals frame-by-frame what the subject is doing. The neural network is a kind of dictionary which provides all the information about the person's situations. This figure represents situation seven: praying. The result can be predicted from the number of situations. So, if the graph lines are followed from zero to 600 frames, what will the result be? As explained in the small box, the solid blue line is the output from the neural network and the other coloured lines represent the rest of the various situations. At frame 1, the subject starts appearing in the scene. The solid blue line changes from frame to frame because the person is walking until around frame 180, bending (180-200), lying (200-210), kneeling (210-240), lying (240-250), bending (very short time), standing (very short time), kneeling (290-295), lying (290-295), praying (295-350), kneeling (350-370), bending (370-380), standing (380-405), bending (405-430), lying (430-450) and standing (450-..). All of these sequences of situations indicate that the person is praying and that it is a normal situation. The system is working well, and reveals everything that is happening frame-by-frame throughout the video.

8.5.2 The Identification of the Situation (fall5)



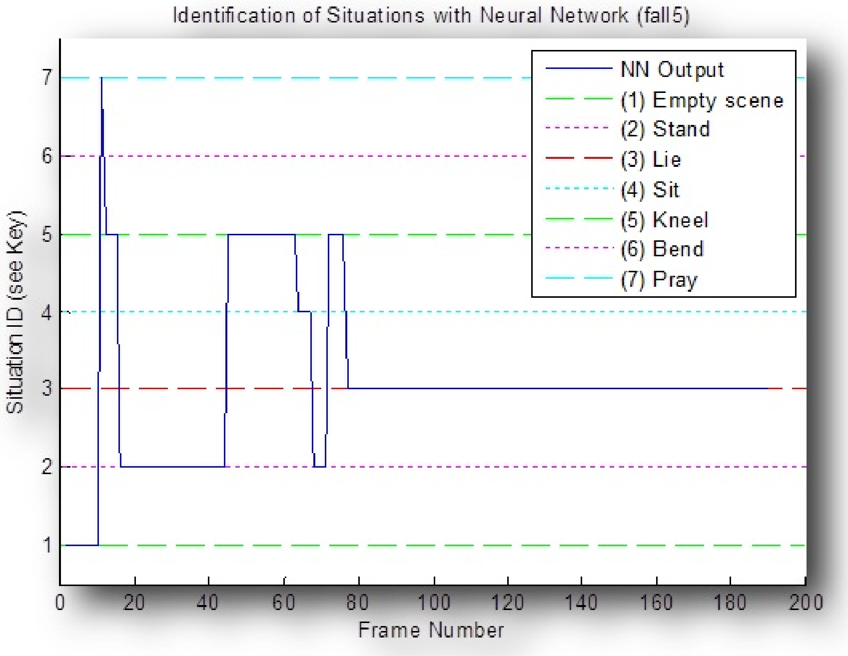


Figure 8.6: Different actions

As shown in the figure above 8.6, the different actions identified during this session are as follows. There is the identification of praying at frame number 15, kneeling at around 40-60, bending at frame 70, and then lying down, which remains the same for the rest of the frames. This means that a fall is detected. Therefore, the fall detection system was able to distinguish a real fall among the various consecutive situations that followed prior to the fall. The constant line after frame 80 showed effectively that the person was lying down. In fact, the number of frames that stay the same for any particular situation is important, as, for example, there is the act of praying (momentarily) at frame 15, then kneeling, which occupies just a couple of frames, then standing, which occupies about 20 frames of the video, followed again by kneeling, sitting and moving up to kneeling again before being categorised as a fall.

8.5.3 Identification of (fall6) and Special Activity (Lying Down)

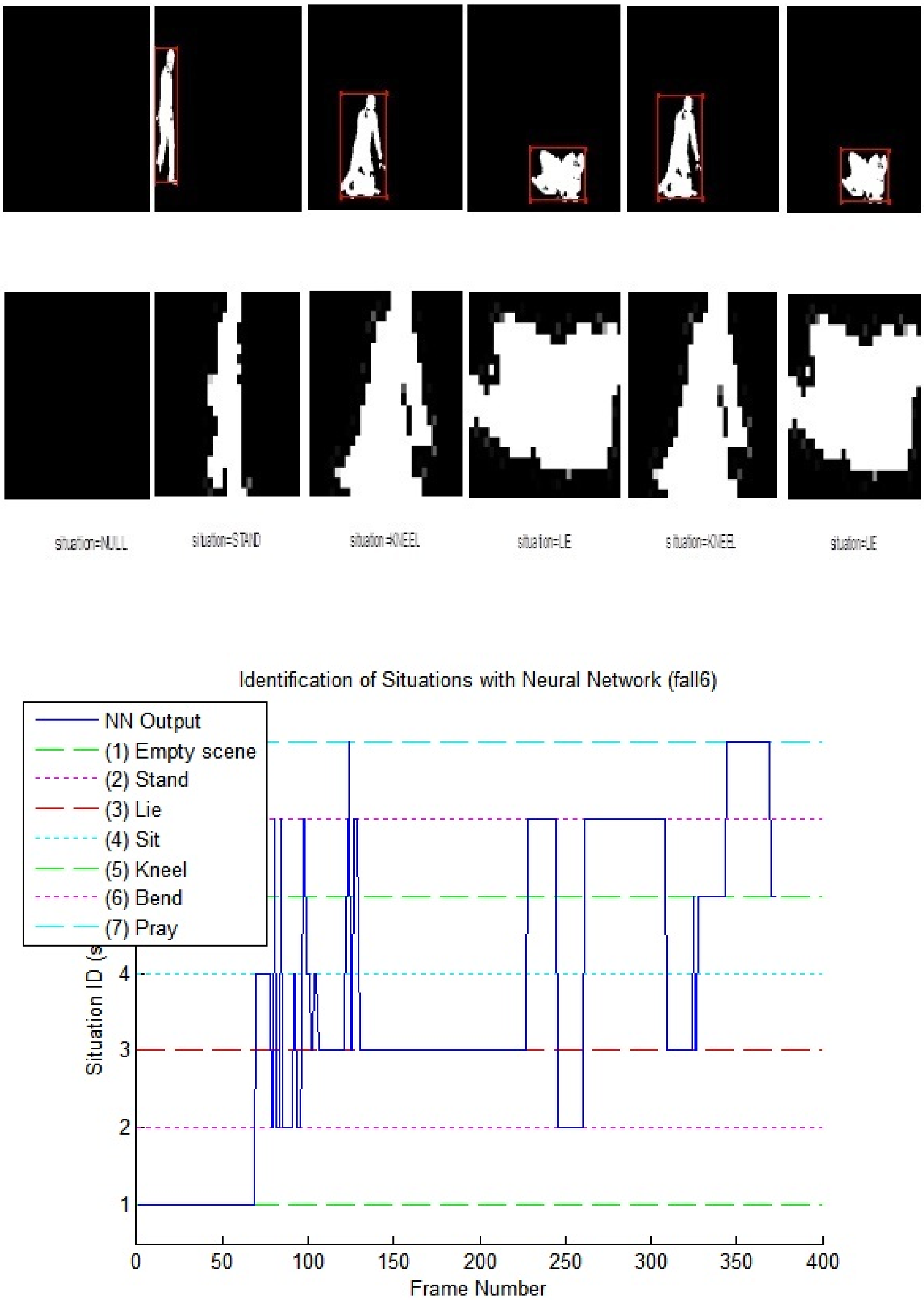


Figure 8.7: Identification of (fall6) and Special Activity (Lying Down)

The graph in (Figure 8.7) represents a movie (fall6.avi) which consists of 600 frames and shows the participants' movements while falling, frame-by-frame. The neural network remembers the predefined situations and tries to relate all the information about the person's situations to the predefined situations, which are standing, lying down, sitting, kneeling, bending and praying. The solid blue line represents the output from the neural network, and the coloured dotted lines represent the level of the associated different situations. Between frames 50 and 125, several activities were identified while falling down. Then, between frames 125 and 220, the person was in fact identified as having fallen down and was then lying down, as shown by the corresponding red dashed line associated with lying down. The combined monitoring in this case therefore identified the different activities during the act of falling down and once the person was finally lying down. Consequently, this is a fall that was successfully detected by the intelligent neural network algorithm.

Figure 8.8: shows the sequence of falling down

Here, the figures show a falling down sequence. It can clearly be seen that in fact, the act of falling comprises a different situation to that in the previous example; the first figure shows somebody kneeling down, the second shows the person bending with the arms stretched towards the ground, and the next shows them lying down flat. The corresponding silhouettes are shown below the respective situations, as well as the results obtained by the neural network in identifying that these situations correspondingly are kneeling, bending and lying.

8.5.4 Identification of (fall7)

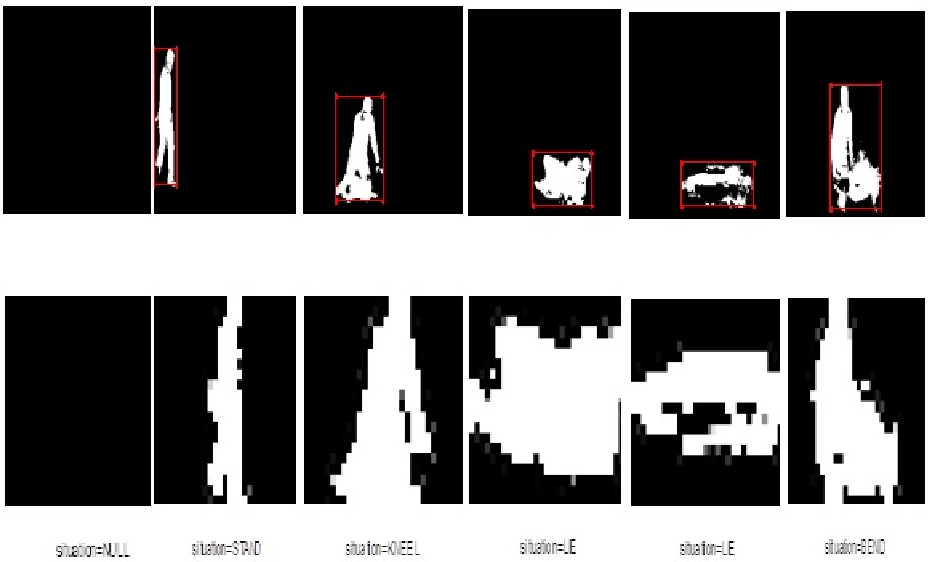


Figure 8.9: shows the sequence of falling down

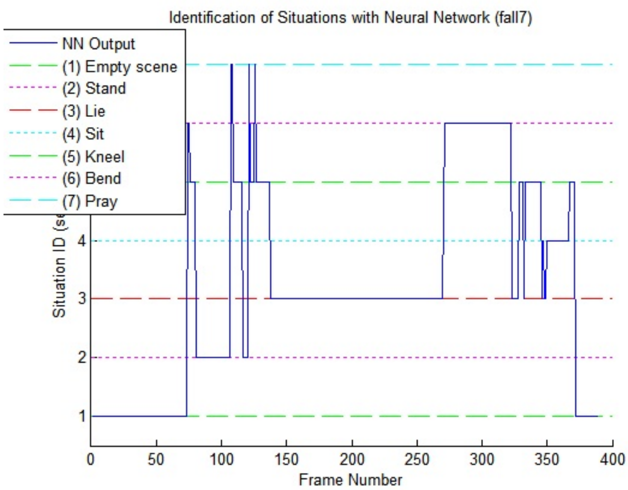


Figure 8.10: shows the sequence of falling down

In the fall7 video, seven situations are identified, as shown by the legends. The first situation detected by the implemented fall algorithm was standing. This is followed by a bending situation, then praying, kneeling, standing, praying, kneeling and lying down, which occupies about 100 frames, after which a fall occurs. Again, this fall scenario reveals that the fall detection algorithm is very sensitive to the changing aspect ratio and the movement of the person in time. The number of frames (or the time for which the person is lying down) is also a crucial factor in determining whether or not this should be categorised as a fall.

8.6 Outlines of the Experiments Performed

8.6.1 Protocol for Data Collection

Healthy participants voluntarily took part in this research study. They each performed different falling or situation scenarios of short durations, and this data was recorded as clips or movies that were stored for processing and analysis purposes using the MATLAB platform and toolboxes.

8.6.2 Tools and Algorithms used for Analysis

MATLAB platform version 7.0 was used for the implementation of the various algorithms of the equations, and these were applied to the recorded clips based on fall scenarios. The video characteristics were 300 x 450 pixels, 30 frames per second, and indeo5 compression was used. It was found that the approximate median filtering gave good results in terms of a compromise between accuracy and speed of processing, which was a couple of seconds. The approximate median filtering performance was compared to a mixture of Gaussians, and it was found that the mixture of Gaussians made the computation more complex and increased the computational time. The suitable threshold was investigated, and this was optimised to a value of 30 pixels, which were used to determine whether a pixel represents a foreground or background.

8.7 Summary

An effective situational approach was used to detect falls among elderly people. This approach is inculcated in the video camera fall detection system using a neural network algorithm, which is easily computed. A mixture of fall metrics was also used to identify the act of falling, in order to ensure that the fall detection algorithm is robust and functioning well. As a whole, the fall detection system using the video-based method is very effective; it is not cumbersome and it easily tracks the silhouette of a person's activity without compromising his or her privacy. In addition, as long as there is a relevant database and predefined situations, the neural network is able to learn new situations. In this chapter, the algorithm codes are elaborated on and discussed to demonstrate the ease of implementing them in real-life situations. However, there are some drawbacks and limitations to this system, and these will be further discussed in the evaluation chapter.

Chapter 9

Case studies of the fall detection system using an intelligent video camera

Objectives

- To assess the neural network algorithm
- To elaborate specific activities with continued monitoring such as lying down and praying
- To test the system using limping and stumbling scenarios

9.1 Introduction

Annually, there are many senior citizens that experience accidental falls which cause physical harm to them as help or support is not brought on the spot of the fall as quick as possible [48]. Therefore, there was a need to devise a reliable and accurate fall detection system that can monitor these unexpected falls so that medical help or support can be provided to the injured person quickly. The aim of this research study is to make sure that elderly people stay active and be as independent as possible. As such, a video fall detection system seems most appropriate in terms of reliability and accuracy as well as preserving the privacy of these elderly people. Here in this case study, the data sets that were used to simulate and optimise the neural network algorithm in detecting a fall are described.

These data sets comprise of both falls or non falls activities that are important to assess the neural network algorithm and also to make sure that a variety of situations are recorded for analysis purposes. Specific activity with continued monitoring such as lying down is elaborated as well as several case studies or scenarios are described.

The first case (Case Study 1: falling/get up) described the scene where a person falls down and then gets up immediately. The challenging parts of this activity is to detect a fall before the person gets up and normally in real life situations that can vary from at least a split of a second. This fall detection system which is developed in this research effectively detects this fall and the on-going activity that follows it.

The second case study (Case Study 2: falling/lying down) describes the scene where a person falls down and does not get up which often occurs among elderly people. Moreover, the difference between falling and lying down will be elaborated.

The third case study (Case Study 3: Sitting) describes the sitting actions or sitting positions of participants and this case was described to distinguish clearly the sitting stance from a lying down stance and how the fall detection system effectively detect this type of situation.

The fourth case study (Case study 4: Praying) is praying which was important to describe here as this involve a series of situations and also to make sure that the fall detection system does not misinterpret falling scenario with praying activity. Here also the fall detection system will be proven to distinguish simultaneously the actions involved during praying.

Lastly but not least was shown some situations of lying down (Case Study 5: lying down) where the participants lied down on several positions in order to distinguish between the act of falling and lie down or just lying down. Last but not least, two further cases showing non-normal gait walking posture ending with falls were shown in Case Study 6 which shows a limping case with fall and case study 7 shows a stumbling case with fall.

The fall detection system, using an intelligent video camera, was able to identify most situations successfully and it seems the best available solution to monitor unexpected falls for the elderly as per the efficiency of the neural network algorithm in detecting the simulated falls as well as the elderly people will like this system as this does not interfere with their privacy.

Students (Fall/Non-Fall/Others)

In order to get reliable data sets, trained and healthy young students were used as participants to do these falling stunts or the various requested non fall activities such as praying, sitting, kneeling and lying down. These participants were required to repeat also the experiments so that reliable data are recorded for the modelling and analysis purposes. The data is reliable when the falling activity is as near as possible to real falls or when the recorded data has successfully covered the desired portion of action or activity to be analysed.

(a) (b) (c)

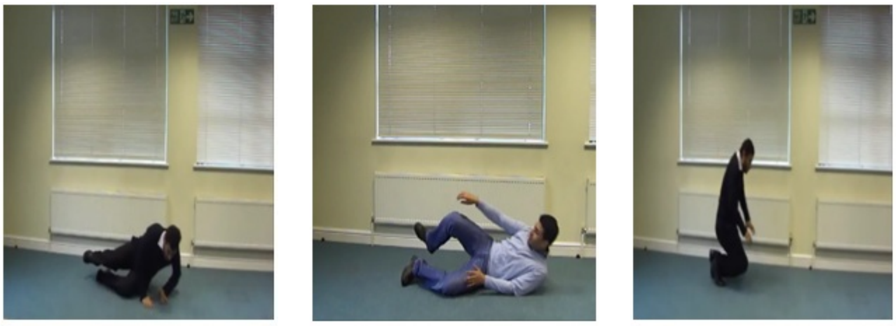


Figure 9.1: Examples Different Types of Falls

There are various types of falls. The first type of fall as shown by the figure 9.1-a in the row is sideways fall, the second is a backward fall as shown in figure 9.1-b and the third is a frontal fall in figure 9.1-c. In all cases, these can cause serious injuries to the elderly person such as broken ribs, injured knees, head or neck. So, these types of falls were evaluated by the fall detection algorithm and they were all identified successfully There are various types of falls. The first type of fall as shown by the first figure in the row is sideways fall, the second as positive falls.

Examples of Non Falls/others

(d) (e) (f)

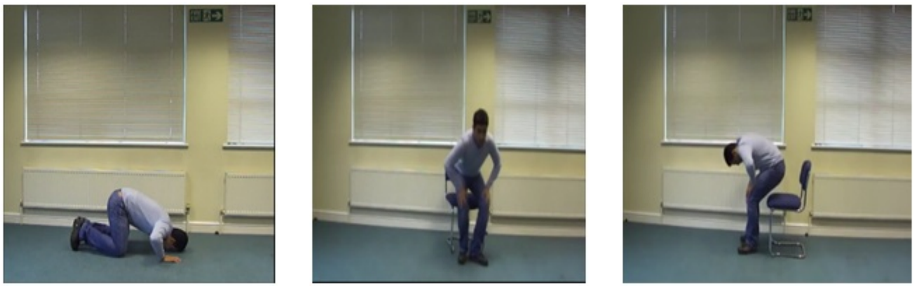


Figure 9.2: Examples of Non Falls/Others

The non falls comprise of the actions of praying, the actions involve in sitting down on a chair, kneeling down, squatting, standing still. The non falls activity are important here in this research in order to assess the implemented neural network algorithm in successfully finding out whether a non-fall action is really a non-fall because if it is the contrary, then the algorithm fails to identify the correct activity of the person. So the participants have performed the non-fall activities well enough so that the neural network algorithm can identify these types of actions as non-fall. Moreover, an empty screen (black screen) is also categorised as a non-fall item.

9.2 Data Sets

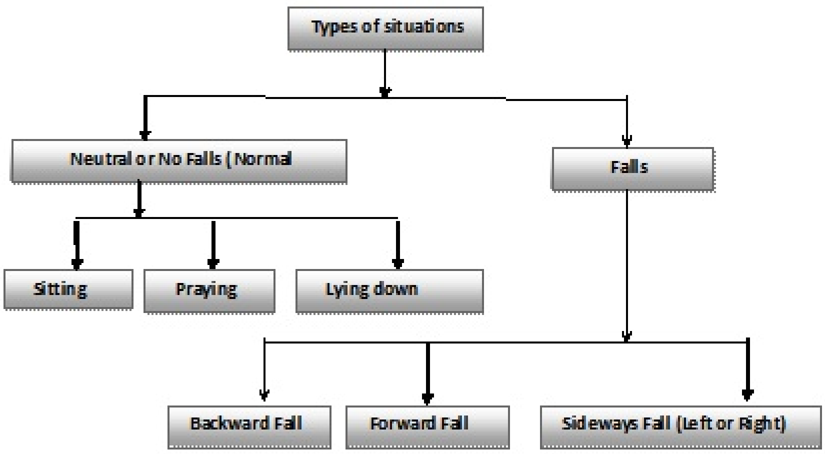
Characteristics of the video clips	
Video original size (average) and type	300 × 450 RGB
Recorded time for each data video	10 to 20 seconds
Video frame rate	30 fps
Video compression method	Indeo5

Table 9.1: Description of the Type and Characteristics of the Video Clips or Data Used for the Fall Detection System.

The recorded data were in the form of small duration (10 to 20 seconds) video of original size about 300 x 450 RGB and the compression method was Indeo5 and the video frame rate was set at 30 fps. The data sets are composed of different falling, sitting and praying scenarios. The data sets are named in terms of categories that are falling (fall5, fall6, fall7) and then praying scenarios that are pray5, pray6, pray7 and sitting scenarios that are sit5, sit6 and sit7. It is worth to note that these data can comprise several situations such as the act of praying (Muslim) which involve a series of situations which involve kneeling, bending, and placing the head on the ground for a certain period of time, sitting up followed subsequently with a standing up position. These data sets are then processed so that the respective silhouettes of the person's activities are extracted from these raw video data. These silhouettes will allow the researcher to be able to use it efficiently in the learning algorithm and also less bulky in terms of size for use in the neural network programming algorithm.

The various activities that were performed by the participants to assess the fall detection system are illustrated by the following set of figures 9.3. The architecture is divided first into two groups of falls and no falls that are subsequently divided into subcategories that make up the database used in this research study. These figures show a range of everyday situations.

Figure 9.3: Diagram of the Classification of the Database Used to Analyse and Model the Fall Detection System (Video-based).



(a) (b)

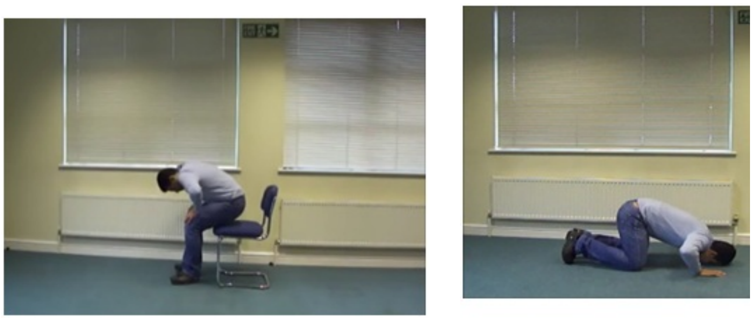


Figure 9.4: Examples of Sitting and Praying

The first situation as shown in Figure 9.4-a shows the participant performing a sitting action on a chair with the head and the figure besides it Fig. 9.4-b shows the stance of praying action with the head placed on the floor.

(c) (d)

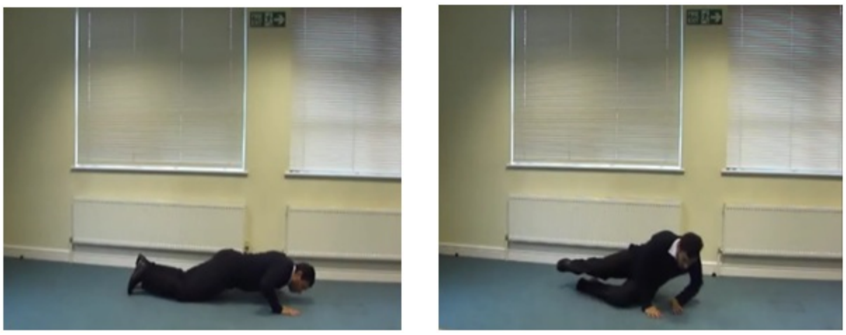


Figure 9.5: Examples Falling Forward and Falling Sideways(early stage)

Furthermore, it is clearly observed as shown in Figure 9.5-c the stance position of a person (stunt-participant) falling flat down on the frontal side of his body. The next figure 9.5-d shows an earlier situation of a sideways fall due to a lateral rotation of the participants'legs.

(e) (f)

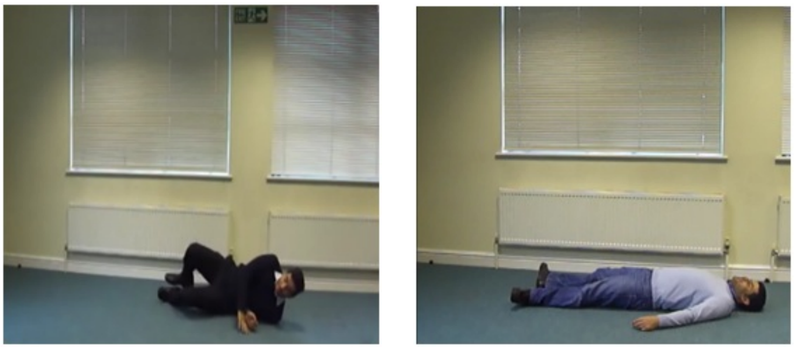


Figure 9.6: Other Examples of Falling Sideways and Lying down on the Back

Figure 9.6-e shows a later stage of the sideways fall event that occurred as shown in Figure 9.5-d with the body fully rested on sideways position. Figure 9.4-f shows the participant lying flat down on his back.

(g) (h)

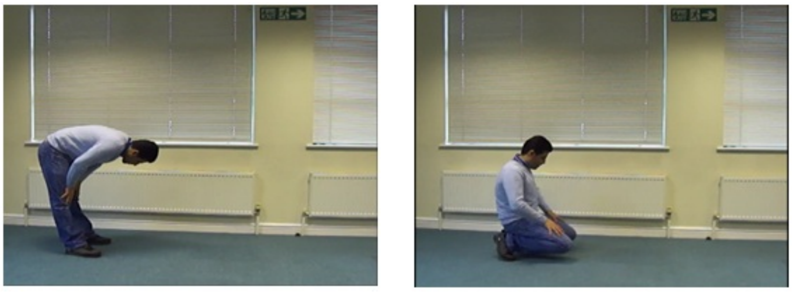


Figure 9.7: Examples of Bending Forward and Kneeling Down

As shown in Fig.9.7-g, the participant is bending forward with the head pointing to the ground and hands pressing on the knees. The second situation (Fig 9.7-h is the act of kneeling passing through a brief squatting mode to effectively kneeling down. Both form part of the common events that take place during the praying situation. Therefore, various activities were performed by the students or participants to have a large range of reliable data and also to assess the neural network algorithm. Moreover, these large database of data consisting of various activities such sitting down, praying, bending down while kneeling with head down, lying flat down on the belly, or on the side (on the ribs), or lying on the back or bending down while standing or squatting all are good situations for the algorithm to learn, adapt and also to identify new situations and their associations whether lying down, sitting or praying for instance. Without such range of data, the analysis of the algorithm may not be reliable. As such, one of the future objectives of this intelligent video based camera fall detection system should be able to identify new situations.

9.2.1 Types of data sets for falls (fall5, fall6 and fall7)

The following figures 9.8-a and 9.8-b are the sequence of falling events that were taken from the Fall5.avi video. The sequence as shown in figure 9.8-a happened just before the sequence as appeared in Figure 9.8-b where the participant was flat down on the mat. These sequences display the falls of a healthy and voluntary participant (stunt) on a safety mat to minimise any falling injuries. The participant had made every effort to show as near as possible to a real fall by letting the body fall under gravity. This fall actually occurred by a lateral rotation of the participant's legs while walking on the mat and this subsequently induce this fall.

(a) (b) (c) (d)

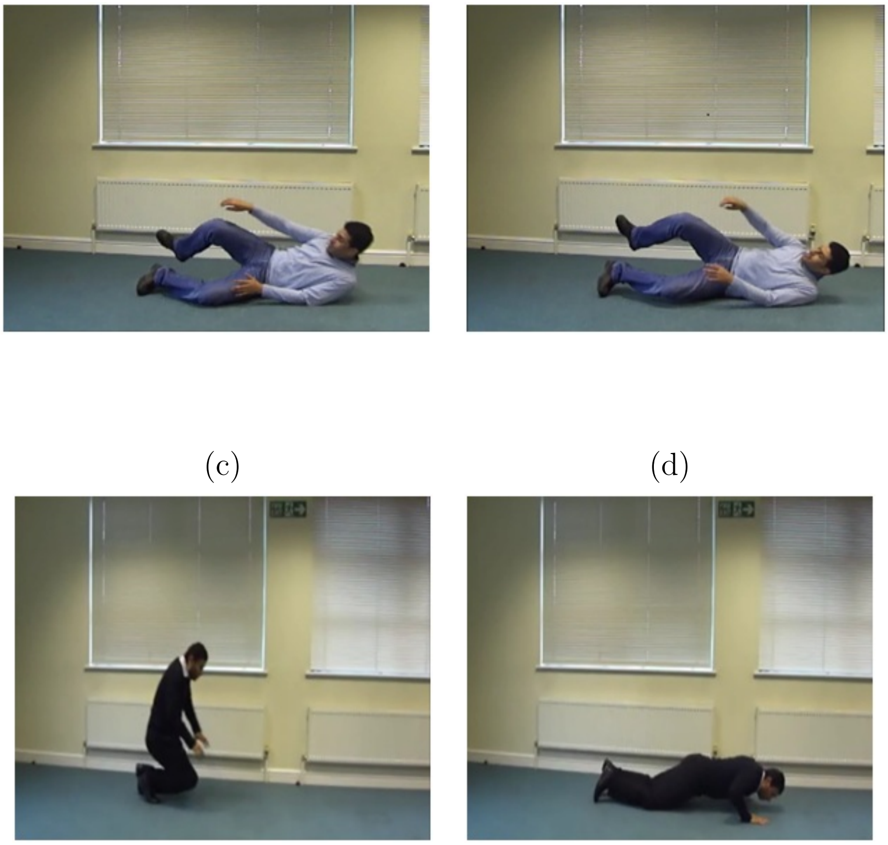


Figure 9.8: Types of Data Sets for Falls

The next type of fall (Fall6.avi), as shown in Figure 9.8-c and 9.8-d, depicts a frontal fall where the participant falls and lies eventually on the belly. This type of fall occurred when the participant was walking on the mat with a sudden frontal force which dis-balance the participant (stunt) which eventually cause the person to fall in a forward direction as shown in Figure 9.8-c until he lies flat down on the belly as shown in Figure 9.8-d.

(e) (f)

Figure 9.9: Other Types of Data Sets for Falls

The third row of figures (Fall7.avi) shows a side fall where the person falls finally on the ribs. This is crucial to have a database of different types of falls as the act of falling down can occur through different ways and this should be known and identified by the algorithm for it to be really reliable fall detection system.

9.2.2 Types of data sets for sits

(a) (b)

Figure 9.10: Types of Data Sets for Sits

The figures (sit5.avi) show the participant in the process of sitting on a chair and how he proceeds of performing the sitting actions with first a bending at the hip sides and then sit down on the chair.

The figure 9.10-a (Sit6.avi) shows a sitting stance of a particular participant by bending the hip which represents a later stage and figure 9.10-b represents an earlier stage of sitting down and the hands place on the knee as a physical behaviour of going to sit.

(c) (d)

Figure 9.11: Other Types of Data Sets for Sits

The figure 9.11-c shows a sitting stance of a particular participant by bending the hip which represents a stage of sitting action and figure 9.11-d represents a later stage of sitting by the bending more the hip but the head is not tilted as figure 9.10-b and the hands place on the knee acts as a kind of support to move down and sit on the chair.

(e) (f)

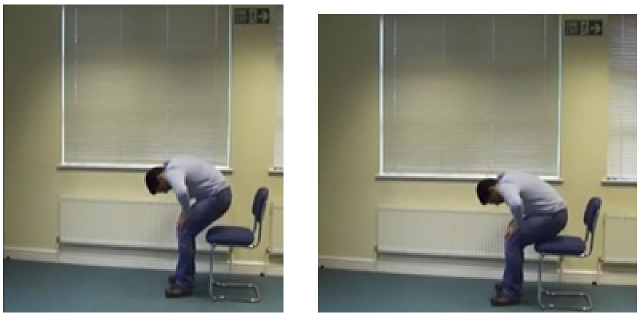


Figure 9.12: Types of Data Sets for Sits

The third row of figures (sit7.avi) shows the profile of the participants while sitting down on a chair. In summary, the sitting actions involved bending the hip, with either the head tilted or non-tilted with the hands lock to the knee acting as support.

One can observe how the aspect ratio (the ratio of the length to the width of the person's silhouette) changes while bending slowly up to a point where the participant is in fact sitting in the chair. This sitting action comprise still of different body postures that may change the aspect ratio and so the algorithm should be able to decipher all this change and acknowledge it as a sitting action.

9.2.3 Types of data sets for pray (pray5, pray6, pray7)

(a) (b)

Figure 9.13: Types of Data Sets for Pray

The praying situation involves a series of physical activities. One activity as shown in Figure 9.13-a is kneeling with the hands placed on the thigh on the front side and the body straight and toes touching the ground. In figure 9.13-b, another physical stance is the prayer kneeling down, and bending the hip and placing the head down on the ground and the hands are placed on the ground that act as a body support.

(c) (d)

Figure 9.14: Other Types of data sets for pray

Figure 9.14-c shows the prayer bowing down (head placed on the floor) with the

hip more bent than Figure 9.13-b to make sure that possible real life aspect ratios are determined for such praying stance. In figure 9.14-d represents an early stage of praying with the prayer going for an act of bending the knee and the hip with the hands placed on the thigh. The head is also pointing down.

(e) (f)

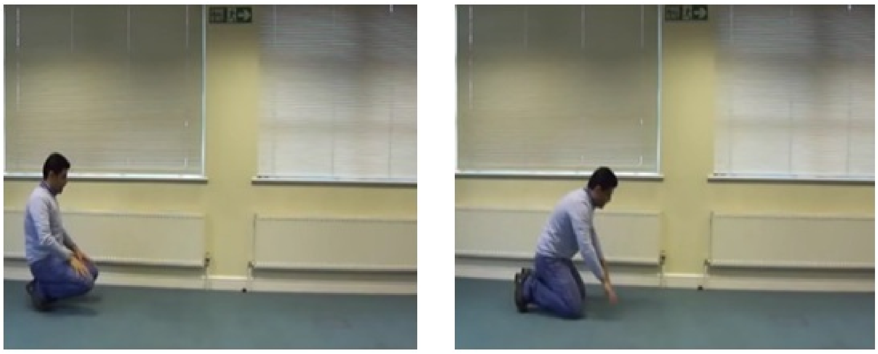


Figure 9.15: Other Types of data sets for pray

The act of praying in fact involves many actions or situations such as kneeling and bending as shown in the first row of the above figures where the person is in the process of praying by kneeling down and then placing the head down, and so the aspect ratio of the respective silhouette will eventually decrease which will trigger different respective predefined situations of the neural network algorithm. The second row is the same participant but this time, he is bending down in front of an object which is the chair as sometimes, in the background there may be more than one still objects or inanimate objects around the person. Therefore, it is good to test the neural network algorithm in identifying the person in such cases where there is the presence of obstacles or inanimate objects. The third row shows the various poses of the person while in the process of praying and this kneeling motion from backward to front can in fact change the aspect ratio drastically and may make it tough to identify it as praying but the video camera was able to identify this as a praying activity.

9.3 Case studies

9.3.1 Case Study 1 (Falls and Does not get up)

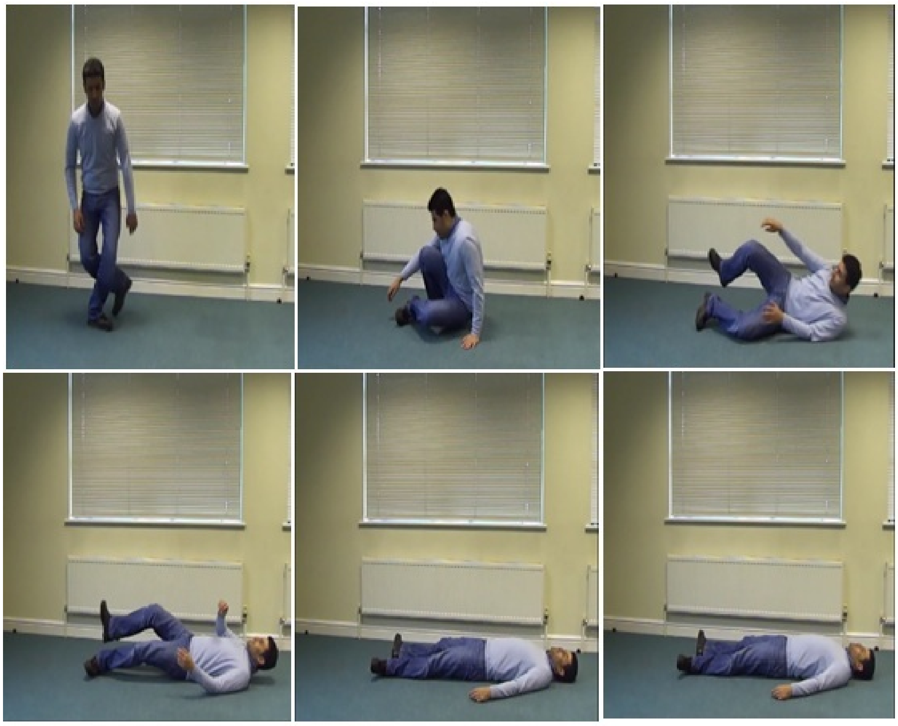


Figure 9.16: Case Study 1 (Falls and Does not get up)

In order to distinguish between falling and lying down, two scenarios are displayed here to clarify these two activities and also these two scenes show one person falls down and then lies down for a while and the second scene shows one person actually falls down and then gets up immediately. The first case study (case 1) shows a person experiencing a lateral rotation of the knees and this cause the person to fall down as shown by the first row of sequence of the video (fall5.avi). The person fall backwards and then continues to lie down. The first problem of such types of fall is that it may involve several activities (like rotation of the legs and falling on the back with the hands facing down, lying on the ribs or on the back). The video fall detection system was able to detect this as a fall even though there was quite a sequence of activities that occurred before this fall.

9.3.2 Case Study 2 (Falls and gets up)

Figure 9.17: Case Study 2 (Falls and gets up)

However, in case 2, another participant, while walking suddenly falls forwards which make him to fall on the frontal sides of his body. After falling down, the person immediately gets back onto his feet and continues his walking motion. The problems in detecting such falls are that the detection system should be able to find out this fall by analysing the perpendicular sideways motion, the sequence of physical stances that happened before falling down and also what motion happen when recovering from a fall and standing up.

9.3.3 Case Study 3 (Sitting)

The case study of presenting a sitting stance was important to analyse because sitting involves the bending of the hip and the range of motion of the upper limbs as well as the bending of the lower limbs such as the knee. The problem with sitting stance is that this can camouflage the activities (bending and falling down like in a real fall) and this was why it was important to show this here. The fall detection system was able to detect a sitting situation and can distinguish this from a fall stance.

Figure 9.18: Case Study 3 (Sitting)

9.3.4 Case Study 4 (Praying)

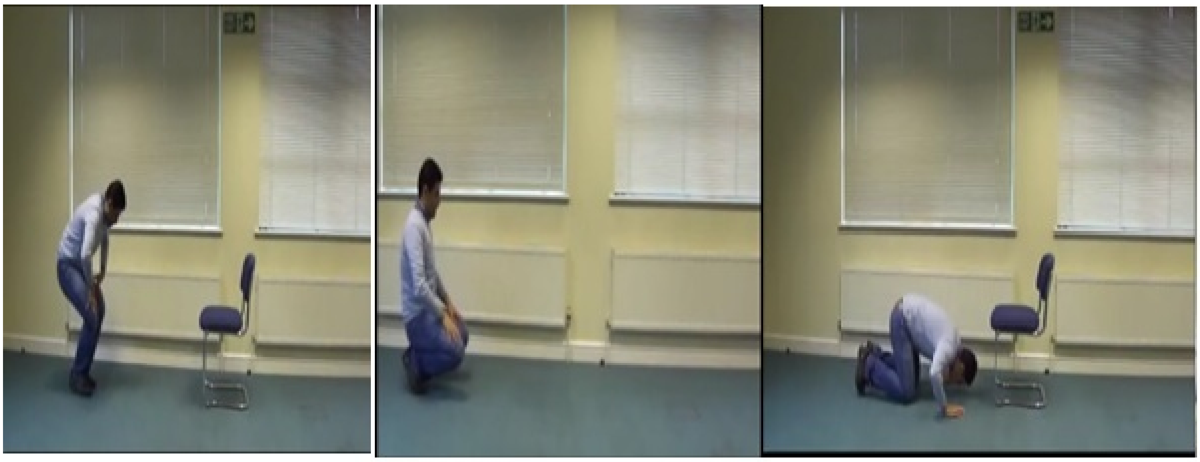


Figure 9.19: Case Study 4 (Praying)

Praying situation involve a range of physical activities and it can comprise overlapping activities as observed for real falls. Praying situation can involve bending the knee, the elbow, bending down the head and placing the head on the ground while kneeling and then kneeling and sitting up straight. It was deemed important to analyse the fall detection system in detecting whether praying is a no fall or a fall and moreover, praying action is a very common activity. As one would not like to have erroneous alarms, the video-based fall detection system was tested and was able to decipher all these praying stance activities and categorised these as no falls.

9.3.5 Case Study 5 (Lying down)

Figure 9.20: Case Study 5 (Lying down)

Finally, just lying down cannot be categorised as a fall. A fall happens when the physical body in a sitting or standing position loses its balance and falls down. Therefore, if a physical body whose aspect ratio was different at an initial frame of reference and this changed abruptly to an aspect ratio corresponding to lying down, then this is categorised as a fall or else this should be categorised as a no-fall activity.

9.3.6 Case study 6 (Limping)

The fall detection system was also tested using limping scenario. It was performed by stunt falling participants to see if the fall detection system that was devised can really detect real-life situation falls such as crippled elderly people of discrepancies in their walking gait or posture.

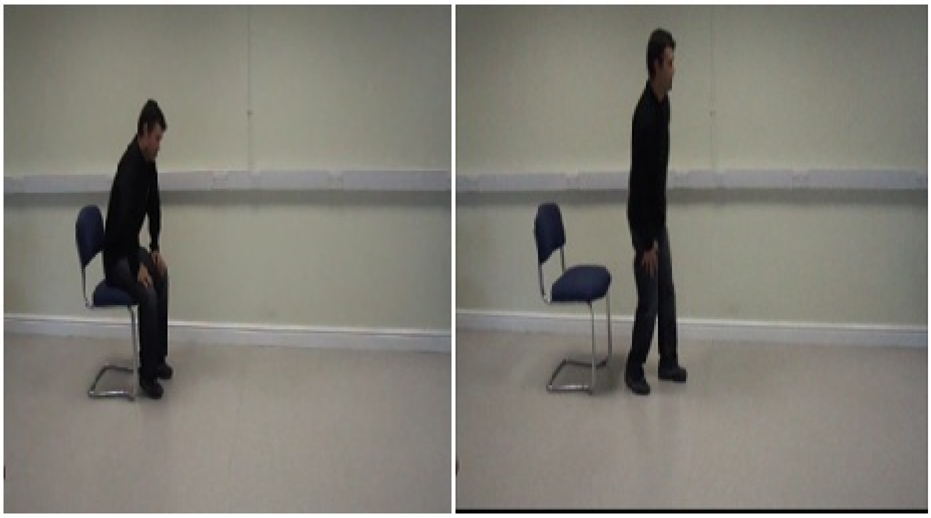


Figure 9.21: Case study 6 (Limping)

The first two selected frames (as shown above) of the limping video shows the person who will be performing the stunt fall moving from a sitting posture to start walking. In so doing, he limps as he moves forward. However, in the following figures, he experienced a sudden fall which was caused when he lost his body balance while walking using his disabled walking gait posture. He fell down in a forward manner and then falls on his chest on the floor. Then in the last figure he stands up and continues his motion.

Figure 9.22: Other Testing Using Limping Scenario

9.3.7 Case Study 7 (Stumbling)

Furthermore, the fall detection system was tested using the stumbling scenario. This was performed by stunt falling participants to see if the fall detection system could detect real-life situation falls that occurred among the elderly people.

Figure 9.23: Sequence of Events for the Case of Stumbling

As shown in the figure above, the stunt participant performing a stumbling motion and subsequent falls. In the initial frames of the video (1^{st} figure top left), the person is walking with a non-normal gait posture and while walking towards the chair (2^{nd}) figure top right), he stumbles (3^{rd} figure bottom left) as shown by the falling backward motion and then he falls flat down on the floor (4^{th}) figure bottom right).

9.4 Summary

The motivation behind this research project is the apparent increase in the number of accidental falls which make it one of prime objectives to research on suitable fall detection system that can be not only reliable, accurate and cheap but also a system that preserve the privacy of the person. Therefore the case studies presented here are based on a designed and tested fall detection system using a smart video camera that makes use of a neural network algorithm that learns and identifies specific activities of the individual in real time. The participants who took part in this study performed falling stunts as well as other non-fall activities in order to have reliable and enough data to implement an accurate video-based situation identification system. It was observed that the fall detection system successfully identifies non-fall as well as fall activities performed by these trained and healthy participants. It was also shown that specific activity such as praying can comprise different postures and hence trigger several situational activities such as bending, kneeling, standing and sitting. Therefore, this video based fall detection system can in fact allow the elderly people to be more mobile and functionally active in their daily lives and should not fear of falling down.

Chapter 10

Evaluation of the fall detection system

Objectives

- Comparing between video-based and portable fall detection sensors
- Evaluating of the neural network algorithm
- Evaluating of the algorithm codes and alarm generation
- Comparing with existing graphical models (HMMs)
- Evaluating of results
- Comparing of NN performance to fall angle, bounding box, HMM and combination performance
- Testing the fall detection algorithms on limping and stumbling ”stunt participants”
- presenting limitations and drawbacks of the current implemented video-based system

10.1 Introduction

This evaluation chapter begins by detailing the advantages and disadvantages of using a video-based fall detection system in comparison with non-video methods. The neural network algorithm and the whole neural-based system as implemented are evaluated, using test codes as well as statistical measures to compare their performance to that of other methods, such as the bounding box, fall angle, *HMMs* and their combined use. A description is also given of the algorithm developed for the video-based fall detection system, which was also tested using limping and stumbling "stunt participants", who were video-recorded to generate data frames in order to test the robustness and efficiency of the algorithm in distinguishing between real falls and non-fall events. The chapter closes by discussing the limitations of the fall detection system developed in this study.

The fall detection system as implemented in this research is evaluated here in terms of its efficiency in detecting a fall, as well as the performance of the algorithm, such as the amount of time that elapses before an alarm sounds. The data collected for the study show that the fall detection system achieved great accuracy and efficiency in detecting falls, and that this situation-based detection system can achieve great accuracy even after being trained using some samples of data (frames). Currently, there are no consistent and rigorous methods detailed in the literature to compare the performances of different fall sensors. However, Noury [128] argues that some common criteria should be introduced and used for evaluation in order to enable the performance of a fall detection system to be determined, despite different procedures being carried out to conduct these fall tests.

10.2 Comparison Between Video-based and Portable Fall Detection Sensors

One of the existing issues with current portable fall detection sensors is that it can take quite a long time before the necessary help arrives at the spot where a fall actually occurs. This delay in summoning the required support or medical treatment can eventually cost the lives of injured people. This is the reason behind the drive to develop a reliable video-based fall detection sensor that will eventually outperform the existing portable fall detection sensors in several ways:

- Compared to portable fall sensors, the video-based sensor is cheaper and more accurate in detecting falls.
- It provides a rapid response to the detection of a fall, reducing the delay in providing the required support to the injured person.
- It preserves the privacy of the data, as only the person's silhouette (a binary image of the person) is utilised.
- It removes the inconvenience for elderly people of having to remember to wear an uncomfortable portable fall detection sensor everywhere they go.

Another method was the design and implementation of SmartFall, which is an automatic fall detection system based on multi-stage thresholding. Mars [129] claim that their SmartFall system achieved a near-perfect fall detection rate for the four types of fall tested in the experiment that they conducted and that after extending the algorithms, together with an eventual assessment of the peak impact force, they successfully reduced the false-positive rate of the system close to zero for all six non-falling activities on which the system was also tested.

Many existing fall detection systems are based on the thresholding method, which continuously compares raw and transformed sensor data with predefined thresholds [55, 130, 131]. In the work of these researchers, a multi-stage thresholding system was used, i.e. one with more than one threshold, all of which had to be exceeded for an alarm to be triggered. Therefore, this system was good in terms of its low false-positive rate. Its drawback was that users must have with them a 'smart cane', which may be cumbersome and not always user-friendly, and which elderly people may forget to keep with them. Thus, the system does not seem to be a practical solution.

Another technique, used by Degen [130], was a triaxial accelerometer incorporated into a watch and attached to the wrist; this used a three-stage acceleration-impact-inactivity algorithm to detect falls by the wearer. While this system did have a low false-positive rate, its detection rate diminished when there was a change in the direction of fall, or when there was significant rotation during the fall. A similar method, developed by Hwang [131] and tested on stunt participants, comprised a sensor box attached to the chest of the wearer. In addition to the accelerometer, these researchers also used a gyroscope and tilt sensor to make the system more accurate. Bourke [55] demonstrated that falls can be distinguished from daily activities by developing a high and low threshold on the angular velocity or the peak acceleration gathered from the sensors attached to the body. They claim a detection rate of 100% for their systems in four categories: forward, backward, sideways and free fall.

However, the participants had to either grip their cane throughout the falling process or release the cane at any time. A major drawback of the SmartFall method are the difficulty in distinguishing between laying the cane down flat and a real fall, because it was found that these two actions had the same profile for normal acceleration. The only difference was the large impact deceleration produced in a fall.

As part of the evaluation of the video-based fall detection system, camera calibration is an important factor, as this is correlated with the body shape change analysis. Camera calibration is also important in improving the performance of algorithms. It is additionally worth knowing that the shape of the body is determined by its posture, its location in the recorded image and the camera projection matrix. It is necessary to obtain a benchmark video so that this can be standardised to identify fall events.

Most current fall detection algorithms were not designed to be flexible and to successfully detect falls by taking account of all the various types of falls; there is uncertainty as to whether a generic fall detection algorithm can be created for all such events. There are issues with detection in low light conditions such as at night; these can be bypassed by using dim light so that this does not affect the users when they are asleep; therefore, efforts should be focused on developing a fall detection algorithm for use in low lighting. Another point is that any camera-based system should be able to operate easily and with minimum setup.

The following table was summerised the properties of each fall detection methods where Itru and R/V/V are intrusion and remote visual verification respectively:

Approach	Class	Property					
		Price	Intru	Accuracy	Live	Setting	R/V/V
Wearable Device	Posture Device	Cheap	Yes	No	Yes	Easy	No
	Motion Device	Cheap	Yes	No	Yes	Easy	No
Installed device	Presence device	Cheap/medium	No	No	Yes	Easy	No
	Posture device	Cheap/medium	No	No	Yes	Easy	No
Camera-based device	Inactivity detection	Medium	No	Depends	Yes	Medium	Yes
	Shape change analysis	Medium	No	Depends	Depends	Depends	Yes
	3D head motion analysis	Medium	No	Depends	Depends	Medium	Yes
Author's video-based fall detection method	Posture device	Cheap	No	High	Yes	Easy	Yes
	Inactivity detection	Cheap	No	High	Yes	Easy	Yes

Table 10.1: Properties of classes of fall detection systems.

10.3 Evaluation of the Neural Network Algorithm

The trained neural network algorithm used in this research study was able to identify the sequence of activities for each recorded scenario of falling, praying and sitting with great accuracy, which will be quantitatively elaborated on in Section 9.5. For instance, for the recorded clips that consisted of falling (fall5, fall6 and fall7), sitting (sit5, sit6 and sit7) and praying (pray5, pray6 and pray7), most relevant sequences of activities were identified for each scene, such as standing, lying down, sitting, kneeling, bending and praying. In addition, during the training of the neural network, fewer than ten iterations were needed to find the coefficients or weights of the neural network, and these were based on 31 frames only, which were sufficient for the building of this neural network. With these few frames, the neural network proved to be efficient in distinguishing a fall or non-fall activity scenario, as well as the sequence of events that occurred during the recording of any particular scenario (pray, sit or fall). Therefore, the neural network implemented in this research was successful in the identification of most situations and produced the corresponding situation vectors for identification purposes.

10.4 Evaluation of the Algorithm Codes and Alarm Generation

The computation of the overall algorithm took on average no more than 12 seconds to identify a non-fall or fall activity. The codes are not complex to implement; therefore, no expensive hardware or software is required for the concretisation of such a video-based fall detection system. As a mixture of Gaussians made the computation of the silhouette extraction more complex and increased the computation time, an approximate median filtering was used, which performed better and had a shorter computation time. The silhouette was resized to 20 x 30 pixels so that, in all, 600 data values acted as input to the neural network. Carrying out this resizing function may have introduced some biases, such as the amplification of some artefacts, or misleading aspect ratios of the bounding box. However, the bounding of the silhouette and its repositioning to the centre of the greyscale image worked well. When the respective vector was assigned to each situation, the trained neural network could easily identify a fall or non-fall activity. Regarding the alarm generation algorithm, the initial alarm count was set to 90 frames and all the predefined situations were introduced into the alarm generation codes. In all, there were three states: STATE_NULL, which was set at 0, STATE_LYING, which was set at 1, and all other states,referred to as STATE_ALL_OTHERS, which were set at 2. An alarm was initiated only when the counter was greater than 90 frames,i.e. three seconds, as the frame rate was set to 30 frames per second. In fact, when the counter exceeded the preset 90 frames, this gave a very accurate measurement for the sample of data; this could be adjusted according to the mean of a larger population size, and especially for elderly people. The sound alarm was included in order to attract the attention of people nearby or passers-by.

10.5 Comparisons with Existing Graphical Models (HMMs)

A popular approach to modelling and inferring human activity is the use of probabilistic graphical models [132]. This comprises the dynamic Bayesian networks also known as *HMMs* [63, 133]and their variants (hierarchical, entropic and coupled *HMMs*). HMMs involve many computations [127] and use the Baum-Welch algorithm for the learning task, which is an example of expectation-maximisation. The model likelihood is determined using the forward-backward procedure (a dynamic programming algorithm). Classic *HMMs* represent any particular state using a single discrete random variable, while a dynamic Bayesian network *(DBN)* represents a state using a set of random variables. Murphy [134] showed that a DBN representation of an HMM may be used in order to reduce the computational complexity of inference by better addressing the need to use minimal data for training the algorithm model. However, all these complex computational issues were spared by the coding for this system, as the neural network seemed to outperform existing models in terms of performance, efficiency and the rapidity of the learning process, using a few samples of input data. The neural network is a popular pattern-recognition algorithm and is based on a heuristic, as well as involving less computation, as observed in this research, where the neural network was found to be a suitable technique for activity recognition because it offers reliable identification of fall situations. To reiterate, the neural network developed here was able to identify the relevant situations with just a few iterations of the algorithm and consisted of only one hidden layer, hence representing a simple architecture for implementation purposes. The neural network was able to learn new predefined situations during the process and therefore did not require the complete replacement of the technology installed for this video-based fall detection system.

Anderson and his colleagues used a voxel space for the evaluation of a video-based fall recognition system for elderly people, utilising three measures to evaluate its performance, as well as the successes and difficulties of the video-based fall detection method [99]. Their system was based on soft computing and the activities were perceived using a linguistic summarisation of activity approach. The researchers used a robust method to build up a three-dimensional object of a representation of the human body, based on the back projection of silhouettes from multiple cameras viewing a given scene. The recording environment was partitioned into discrete regions, called voxels. The camera then constructed a list of voxels that intersected with the viewing volume; only the pixels from which this voxel was viewable were recorded. A fall dataset comprising 18 sequences was used. The results demonstrate that the system captured all on-the-chair as well as all on-the-couch states, representing activities that depend on the spatial location in the room of the voxel person, together with the static object.

Performance was not very good, in that the system was able to classify only 82.7% of upright states; some of the time intervals (16.9%) were also classified as on-the-ground. This relatively poor performance was due to the fact that the system was not able to reconstruct the object because of the biased viewing angles. A second issue was caused by the fuzzy sets used to build the learning rules, which were defined by humans. Some situations were a perfect fit to the empirically fuzzy sets and the current system used in this research was able to determine each activity with great accuracy. The authors also acknowledge that it was still necessary to perform some work in order to match the exact number of linguistic summarisations and the hand-labelled activities and frame-by-frame decisions made. Despite all this, they state that their system produced a sufficient number of on-the-ground summaries in classifying falls and was able to distinguish between fall and non-fall activities.

Foroughi [16] suggest a novel approach for human fall detection based on a combination of the eigenspace technique and integrated time motion images. An integrated time motion image is a form of spatio-temporal data recording both motion and time of motion. Application of the eigenspace algorithm to the integrated time motion images results in the extraction of the eigenmotion. Next, a multilayer perceptron neural network is employed to classify the motions and to identify fall events. This fall detection algorithm can also consider a wide range of motions that encompass daily physical activities, suspicious behaviours and unusual events. The system was found to be satisfactory and reliable average of 89.49% in detecting falls.

10.6 Evaluation of Results

The detection of a fall is positive if the fall detection system can distinguish a fall and negative if it cannot. As the output is in binary format, the quality of the detector cannot be evaluated through one single simple test. Therefore, it is crucial to analyse the results using some statistical tests. There are four possibilities in the detection of a fall: a true positive *(TP)* is returned when a fall occurs and the sensor detects it; a false positive *(FP)* is when the fall sensor identifies a fall where in fact a fall did not occur; a true negative *(TN)* is where only normal movement occurs (there is no fall) and the sensor does not detect a fall; and a false negative *(FN)* is returned when a fall occurs but the fall detection system fails to detect it. In order to evaluate the responses in terms of these four possibilities, two criteria are assessed: sensitivity and specificity, which are normally used for evaluation purposes. Sensitivity is the capacity of the fall detection system to detect and declare a fall, while specificity is the capacity to detect only a fall. Equations 4.6.1 and 4.6.2 were used to calculate specificity and sensitivity respectively, while Table 9.2 shows the definitions of *TP, FP, FN and TN*.

Fall incident/System Recognition	Fall occurs	Fall not occurs
Positive	TP	FP
Negative	FN	TN

Table 10.2: Definitions of TP, FP, FN and TN

There are several possible falling scenarios, but a limited number of positive and negative fall situations were used for the evaluation of the fall detection system. Most of the falls that occurred in this research were either in the anteroposterior plane, consisting of both forward and backward slips, or sideways falls. There are many motions in daily life in which the intensity of the movement can be the same as in accidental situations, such as the actions of lying or sitting down quickly, which can imitate falls. Therefore, Noury [128] and Bourke [55] propose a set of scenarios to determine the sensitivity and specificity of a fall algorithm, shown in Table 9.3.

Category	Name	Outcome
Backward fall	Ending lying	Positive
	Ending in lateral position	Positive
	With recovery	Negative
Forward fall	Ending lying flat	Positive
	With rotation, ending in the lateral right position	Positive
	With rotation, ending in the lateral left position	Positive
	With recovery	Negative
Lateral fall to the right	Ending lying flat	Positive
	With recovery	Negative
Lateral fall to the left	Ending lying flat	Positive
	With recovery	Negative
Neutral	Sitting down on a chair	Negative
	Walking a few metres	Negative
	Bending down to collect something and rising up	Negative

Table 10.3: Scenarios for the evaluation of fall detectors

The first column in Table 9.3 represents the fall categories: backward fall, forward fall, lateral fall to the right, lateral fall to the left and neutral (representing a non-fall situation). The second column describes the ending of the fall: whether the person ends lying down flat, on the side, with or without rotation, or recovers from the fall (which means that he or she stands up and continues activity). The third column shows the expected outcome of these fall or non-fall situations, which can be either positive, which means a fall occurs, or negative, which means that no fall occurs. These expected outcomes are then used to evaluate the fall detection algorithm, such that corresponding probabilities for sensitivity and specificity are determined from the expected outcome (theoretical outcomes are normally 100%) and the real outcome from the fall detection algorithm.

In order to evaluate the fall detection algorithm for these evaluation scenarios, one realisation of each scenario per participant was not enough. The participants had to perform several scenarios three times in different orders and were able to rest between tests. Each subject also had to vary his speed in performing any pre-defined scenarios as determined by the researcher. To prevent any learning effect or habituation to the gesture following the consecutive reproduction of the same scenario, the order of the tests was varied. Of the 14 scenarios listed in Table 9.3, half are classified as positive (falls) and half as negative (non-falls). As each subject performed three trials for each condition, then in total each subject did 42 tests, making a grand total of 420 tests for the sample of 10 subjects. These 420 data points are considered sufficient to provide a statistically significant computation of the specificity and sensitivity of the device. It is worth noting here that the subjects ranged in age from 20 to 40 years and were all in good health. It was not possible to sample the target population of elderly people, as this would have involved many additional ethical considerations because of the increased health and safety risks, which would have meant that a medical practitioner would have had to attend the experiments.

Table 9.4 shows the percentage of falls and non-falls detected for each scenario; it was found that a strong majority of the scenarios were successfully classified, with an average of 94.3%. An ideal fall detection system would be able to produce 100% sensitivity and 100% specificity, which can be obtained during the setting up of experiments, but there will be significant loss of performance during the design of an autonomous fall detection system, which may be due to unwanted artefacts in the extracted silhouette.

Category	Name	Outcome(correct detection/ total)	Percentage of Outcome
Backward fall	Ending lying	27/30	90%
	Ending in lateral position	30/30	100%
	With recovery	30/30	100%
Forward fall	Ending lying flat	30/30	100%
	With rotation,ending in the lateral right position	27/30	90%
	With rotation,ending in the lateral left position	29/30	96.7%
	With recovery	30/30	100%
Lateral fall to the right	Ending lying flat	30/30	100%
	With recovery	29/30	96.7%
Lateral fall to the left	Ending lying flat	26/30	86.6%
	With recovery	27/30	90%
Neutral	Sitting down on a chair	26/30	86.6%
	Walking a few metres	30/30	100%
	Bending down to collect something and rising up	26/30	86.6%

Table 10.4: Percentage of detecting a fall or non-fall on normal gait

In total, it was found that there were 199 *TPs*, 11 *FNs*, 197 *TNs* and 13 *FPs* (Table 9.5 and Figure 10.1). These figures were used in equations 4.6.1 and 4.6.2 to give a specificity of 94.8% and a sensitivity of 93.8%. Despite the good overall performance of the fall detection device, these metrics could be improved, reaching 99% to 100%, by taking into consideration various factors such as the physiology and kinematic parameters of the moving individuals or by analysing their gait (way of walking).

Fall incident/System Recognition	Fall occurs	Fall not occurs
Positive	199	13
Negative	11	197

Table 10.5: Evaluation of recognition results

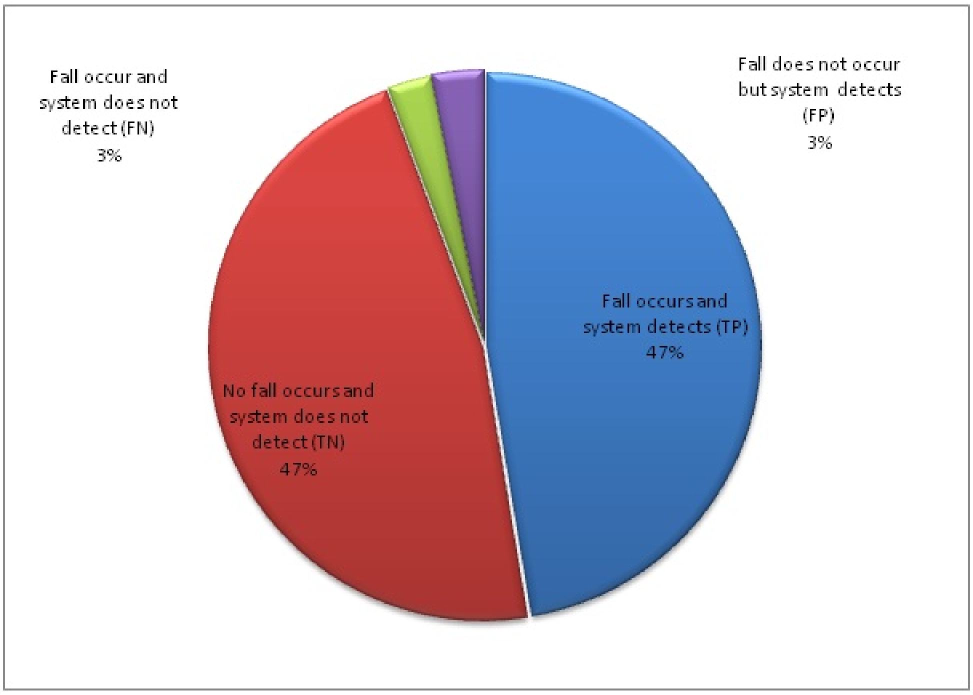


Figure 10.1: Evaluation of recognition results.

10.7 Comparison of Neural Network performance to Fall Angle,Bounding Box,HMM and Combination performance

Statistics tests	NN algorithm	Fall angle	Bounding box	HMM	Combination
Specificity (%)	94.8	81.4	83.3	73.8	89.0
Sensitivity (%)	93.8	82.3	82.8	76.6	84.2

Table 10.6: Comparison of Neural Network performance to Fall Angle, Bounding Box, HMM and Combination performance

When the four fall detection approaches taken were compared (Table 9.6), it was found that while the *NN* algorithm performed best, the algorithm combining the fall angle, bounding box and hidden Markov model was better than any of the three components individually, with a specificity of 89% and sensitivity of 84.2%. The specificity of the fall angle method was 81.4% and its sensitivity 82.3%, while the respective values for the bounding box were 82.8% and 83.3%, and for the hidden Markov model only 76.6% and 73.8%.

10.8 Testing the Fall Detection Algorithms on Limping and Stumbling ”Stunt Participants”

The current fall detection system was tested by including two more scenarios: limping and stumbling. These two types of action were performed by stunt participants so that it could be observed how well the fall detection system could detect scenarios as near as possible to the practical situations of elderly people. The two actions were tested by repeating the procedure above, following the same 14 scenarios, but with the subjects simulating these two handicaps. The results are summarised in Table 9.7. It was found that while the percentage of times the fall detection system detected falls with these two types of handicap included in the walking gait was lower, with a mean percentage accuracy of 90.2%, the detection algorithm was still quite reliable and had a high detection rate.

Category	Name	Outcome(correct detection/ total)	Percentage of Outcome
Backward fall	Ending lying	24/30	80%
	Ending in lateral position	29/30	96.7%
	With recovery	26/30	86.6%
Forward fall	Ending lying flat	27/30	90%
	With rotation,ending in the lateral right position	27/30	90%
	With rotation,ending in the lateral left position	29/30	96.7%
	With recovery	29/30	96.7%
Lateral fall to the right	Ending lying flat	28/30	93.3%
	With recovery	29/30	96.7%
Lateral fall to the left	Ending lying flat	26/30	86.6%
	With recovery	25/30	83.3%
Neutral	Sitting down on a chair	26/30	86.6%
	Walking a few metres	28/30	93.3%
	To bend down to collect something and rise up	26/30	86.6%

Table 10.7: Percentage of falls and non-falls detected with handicapped gait (stumbling and limping)

10.9 Limitations and Drawbacks of the Current Implemented Video-based System

One limitation of the current system is that more robust algorithms should be devised to make sure that the silhouette is noise-free. When the silhouette is noisy, the neural network algorithm may misinterpret the data and produce a false positive. However, in this research, this had no effect on the outcomes of the fall detection system. New methods should also be devised to detect a person falling when walking towards the camera and to avoid a false negative output when the user is carrying a large object.

10.10 Summary

Overall, the fall detection algorithm is efficient, as it is flexible, which means that the number of seconds or the number of counts before sounding an alarm can be predefined. The length of time before sounding an alarm along with the computational time is acceptable, at about 10 seconds. The accuracy and feasibility of the fall detection system were evaluated using statistical metrics and were found to be of an acceptable standard. The implementation of this particular neural network algorithm is cheap because of the simple structure of the neural network, which consists of only one hidden layer; the fast convergence of the coefficients after a few samples of input in detecting a fall makes it very efficient for practical purposes, because this fall detection system can be trained to detect other scenarios and classify them accordingly.

Chapter 11

Conclusion

Objectives

- Summarise our research
- Highlight the original contribution to knowledge
- present the limitations and future work

11.1 Research Summary

This research aimed to investigate and implement a novel, smart and reliable fall detection system in order to reduce accidental falls that occur among senior citizens, especially indoors. It has investigated ways to provide elderly people with good care and to enable them to move more freely. Several image processing stages were defined and described. The RGB image was transferred to greyscale as a first processing stage, after which background subtraction algorithm methods were applied to video-based fall detection. Various tracking methods and feature extraction were discussed in (Chapter 2).

From the literature review, it was found that senior citizens often forget to wear portable sensors to detect falls and that they do not feel at ease when wearing these portable devices, which they find cumbersome and restrictive of their mobility or freedom. To make matters worse, standard portable sensors generate data that is not easy to interpret, with the potential to bias the outcome. This research resulted in the solution of using video-based detection systems, whose advantages are that they allow senior citizens to move freely without encountering any of the issues experienced when wearing portable sensors (Chapter 3).

Image processing techniques such as background modelling and feature extraction types were discussed mathematically in Chapter 4, where the principles of hidden Markov models, neural network algorithms and evaluation metrics were also elaborated upon from a mathematical perspective (Chapter 4).

The neural network developed for this project consists of an input layer, which accepts the data input presented to it, one hidden layer to process the data input and make decisions, and one output layer to display the final solution(s) or outcomes. The training of the neural network represents the core of the neural network algorithm, which employs the supervised learning type of training (Chapter 5).

During the background modelling stage, the foreground of the video image is subtracted from the background image to generate a silhouette which identifies the person as distinct from the background. An approximate median filtering method is then performed on these images, enabling each frame to be compared with the background image; this filtering technique was compared to the outcome obtained when a mixture of Gaussians was used as filters. From this data, the features obtained from these binary maps that is, the silhouettes were further investigated in order to find ways to automate the video-based fall detection algorithm and hence the fall detection system (Chapter 6).

The techniques used in this research to detect falls were grouped into categories, named the first and second approaches. The first approach relied on features of the extracted silhouette. These were (i) the fall angle, representing the angular distance between the ground and the person's centroid, (ii) the bounding box method, where the horizontal and vertical dimensions of a rectangular box that contours the object (silhouette) in motion are utilised to compute the aspect ratio, which provides a mathematical measure to monitor the changing shape of the moving silhouette, and (iii) the hidden Markov model, which was adopted because it uses probability theory to detect changes in the silhouette. A fourth combination method was then used for the first approach, combining the fall angle, bounding box and HMM methods with the aim of overcoming the drawbacks of each used alone. All four fall detection algorithms in the first approach category made use of feature evolution, with time to detect falls (Chapter 7).

The second approach was incorporated into this research with the aim of exceeding the fall detection performance observed with the first approach methods. In this second category, a neural network algorithm was employed in the hope of concretising an intelligent fall detection system with the ability to learn new situations and to identify an individual's specific activities in real time. In fact, the system as implemented was able to identify a predetermined set of situations, as follows: (i) blank screen, (ii) standing, (iii) sitting, (iv) lying, (v) kneeling, (vi) bending, and (vii) praying. A neural network of perceptrons was used to classify each silhouette according to these predefined situations (Chapter 8).

This research has also investigated and evaluated the performance of the fall detection algorithm. The evaluation metrics used were based on statistical measures, such that these measures or metrics could be used to compare the performance of the fall detection algorithm using the first and second approaches. Some limitations and drawbacks of the intelligent video-based fall detection system were also outlined (Chapter 9).

Seven case studies (falling and not getting up, falling and getting up, sitting, praying, lying down, limping and stumbling) were used, based on a tested fall detection system. The individual activities were identified by using a neural network algorithm. Sufficient data was collected as the participants performed fall and non-fall activities. The system successfully identified these and demonstrated that situational activities such as praying, bending, kneeling, sitting, standing and lying were also identified successfully (Chapter 10).

11.2 Contribution to Knowledge

This research has successfully investigated and implemented a video-based fall detection system that meets the needs of senior citizens, especially for indoor applications. The system is user-friendly and will reduce the possibility of senior citizens restricting their movements at home or living in isolation due to fear of falls. Moreover, the fall detection system that was developed preserves users' privacy, even though their daily activities are thoroughly monitored in real time.

Furthermore, the fall detection system that was devised shows promise in terms of high reliability and accuracy in detecting falls (greater than 90%). Therefore, this device can be put into practice without inconvenience and with confidence, removing the tensions currently felt by senior citizens, thereby safeguarding their wellbeing. The system can be manufactured at low cost, requiring minimal installation tools and minimal maintenance, so it can be used worldwide at home or in indoor environments with ease and with minimum supervision. Another advantage of the system developed here is that it identifies different situations or actions of elderly people. It also follows the current trend for environmentally friendly technology, producing no harmful emissions. Therefore, this research has successfully developed a cheap, reliable, accurate and user-friendly fall detection system which preserves the users' privacy.

11.3 Success Criteria Revisited

Camera calibration: The camera was calibrated to monitor the movement of the participants effectively, as well as to analyse changes in body shape through time. In the body shape change analysis, a camera calibration technique improves the performance of the algorithms.

Development of generic fall detection algorithm: Most of the existing algorithms in the literature were designed for one case, without much flexibility. In this research, by contrast, many cases were used with different types of fall, so that a generic fall detection algorithm could be developed for each type of scenario.

Easy setting for camera-based systems: If a camera-based fall detection system is being used, it should be easy to be set up; otherwise users find them hard to accept, because elderly people and patients generally need other people to set systems up for them.

Deployment test: Camera-based fall detection systems should have deployment tests performed on them, and the different systems need to be compared. The deployment test can also be used to collect feedback from users in order to improve such systems.

All the research objectives that were set in this work were met. For instance, when assessed in terms of performance using two different approaches, the implemented fall detector showed promise regarding its acceptance by elderly people, while the technology also proved to be innovative and smart enough to detect falls. The fall detector showed promising results in terms of reliability (regarding both sensitivity and specificity), the combination of the fall angle, bounding box and HMM, the use of a neural network system and the accuracy of the results in detecting falls and non-fall activities; moreover, the fall detector was found to be user-friendly and cost-effective. The study showed how the proposed algorithm outperforms existing systems, thus demonstrating the importance of converting RGB images to greyscale.

The neural network algorithm was selected from among the other methods and included in the fall detection algorithms. A study was made comparing the various advantages of the new fall detection algorithm with existing ones. The problems relating to the position of the silhouette, which can appear anywhere in the picture frame, and concerning the unknown size of the silhouette, were also solved. The importance of the state machine was highlighted, along with how it is depicted and explained, to show the working of the algorithm and the states through time.

During the evaluation stage, the two approaches employed in this research were compared in terms of efficiency and accuracy, and it was found that the neural network outperformed the combination method, along with each individual method used in the first approach. From the literature review and evaluation, it can be observed that opting to travel in the direction of a video-based fall detection system was a good choice.

The neural network algorithm was evaluated in terms of both accuracy and of convergence (less than 10 iterations) of the solutions for the weights of the neural network. The system was assessed for its ability to distinguish falls from non-fall activity in various scenarios. In addition, the algorithm was evaluated in terms of computational time and complexity, as well as the setting of the parameters as a measure of the reliability of the alarm system. The first approach was used to detect falls using the fall angle, bounding box and HMM methods, which were evaluated together with a combination of these methods, as well as the second approach, which involved the neural network algorithm. A wide range of situations was simulated, such as neutral activities or non-fall activities comprising sitting, walking and praying, as well as falling. There was also a wide range of falling scenarios. One type of fall was caused by lateral rotation of the knees, resulting in the stunt-person falling down; others included backward, frontal and sideways falls.

Abnormal gaits were also tested. Specifically, the fall detection algorithms were tested on all 14 scenarios using stunt participants who simulated handicapped gaits comprising limping and stumbling, in order to test the validity of the network algorithm in detecting such abnormal gait movements and categorising them as non-fall or fall activities. This ensured that a wide range of real-life activities prior to falls were covered. The participants performed falling stunts as well as non-fall activities in order to provide sufficient reliable data to implement an accurate video-based situation identification system.

It was shown that the fall detection system successfully identified the non-fall as well as fall activities performed by these healthy, trained participants. It was also shown that specific activities, such as praying, can consist of different postures, and hence can trigger several situational activities, such as bending, kneeling, standing and sitting. Therefore, this video-based fall detection system should in fact allow elderly people to be more mobile and functio8nally active in their daily lives, reducing their fear of falling down.

Unwanted noise in the image was removed using morphological filtering. The silhouette-based and motion-based features were exploited, as they represent important techniques to be employed for fall detection from the system's recorded images.

A neural network algorithm, being an effective situational approach, was used to detect falls among elderly people and was incorporated into the video camera fall detection system, as it can easily be computed. Moreover, a mixture of fall metrics was used to identify the act of falling in order to ensure that the fall detection algorithm was robust and functioned well. The study has shown that a video-based fall detection system is effective in tracking the silhouette of the person's activity without compromising his or her privacy. Additionally, the neural network is capable of learning new situations if presented with various inputs from a database representing particular situations.

It was found that both the visual features and the learning algorithms are important in producing an intelligent and flexible fall detection system.

11.4 Limitations and Future Work

11.4.1 Limitations of the video-based fall detection system as developed

A number of limitations were noted in Chapter 9 (section 9.9), relating to noisy silhouettes and to reduced performance when the person is walking towards the camera or carrying a large object. It should also be noted that in the presence of a handicap such as represented by simulated limping and stumbling, the performance of the fall detection algorithm appears to decline. A final limitation of the experiments was that the findings were obtained by observing young, healthy stunt-participants, rather than the older people for whom it is designed.

Overall, however, the fall detection algorithm has been shown to be efficient, as it is flexible, which means that the number of seconds or the number of counts before sounding an alarm can be predefined. The amount of time elapsing before sounding an alarm, along with the computational time, is acceptable at about 10 seconds. The accuracy and feasibility of the fall detection system have been evaluated using statistical metrics and have also been found to be of an acceptable standard, based on the high percentage of sensitivity and specificity. The implementation of this particular neural network algorithm is cheap because of the simple three-layer structure of the neural network algorithm, which consists of only one hidden layer. Moreover, the fast convergence of the coefficients in detecting a fall after a few samples of input makes it very efficient for practical purposes. This is because this fall detection system can be trained to detect other scenarios and to classify them accordingly.

11.4.2 Future work

It can be argued that the inclusion of experiments involving simulated handicaps created a more realistic old-age walking gait scenario, despite the use of young, healthy volunteer participants. Future work could try to address more handicaps as observed among elderly people, based on their physical movements or the lack of large angular movements of their limbs. One of the main issues was the silhouette segmentation; in the research, it was assumed that there was only one person in the scene or in front of the fall detection camera, due to the independent house environment. Future work could be directed to monitor multiple physical bodies while in motion, aiming to detect multiple falls by people in the camera view. Methods should also be devised to deal with occlusion, which might impede the silhouette profile, consequently affecting the outcome of the various fall detection algorithms.

One possible area of future work based on my current research is to use a multiple cameras voting system to decide whether the overall result or outcome is a fall or not a fall. In addition, this type of multiple camera voting system will involve a fusion stage whether this can be applied early or later. Therefore, future research should put into practice one of the two approaches related to fusion for fall detection based on multiple cameras system. The first approach is called the early fusion approach and the second approach is called the late fusion. For the early fusion approach, the multiple camera views are gathered together to construct the 3D representation of the human body. Semantic features are taken from this 3D representation in order to detect falls. The silhouettes collected from camera 1, camera 2, camera 3 and camera 4 pass through the early fusion stage to build the 3D representation of the Human body. Then the features are extracted in the feature extraction stage which is then followed by fall detection.

The late fusion approach is performed in 2D and each camera automatically decides if a fall has occurred. The silhouettes from camera 1 to camera 4 are gathered and then they all pass individually via a feature extraction stage and fall detection stage and then finally they undergo a late fusion stage whereby a voting system approach is executed to detect falls. Both the early fusion approach and the late fusion approach are then combined to form an overall outcome or the final decision whether a fall has actually occurred or not. As part of ongoing works, future research can also focus on edge detection methods for cleaning the silhouette in the hope to increase the reliability of the fall detection algorithm as it was observed that artefacts (even after cleaning) are not completely removed from the processed images. It is also known that edge detection methods are widespread utilised in the early processing of one-band images that are also referred to as graylevel monochromatic images as compared to multiband coloured multispectral images. Also edge detection is one of the first stages in feature extraction. Future research should not only concentrate their efforts in investigating the log method, sobel operator and canny operator but also morphology operators especially when the log method, sobel operator and canny operator cause noise and discontinuities on the edge of images and also while smoothing images. Therefore, future research should investigate the morphology operators to detect edges and it has been proven to be efficient while being applied to satellite and remote sensing images. The morphology operators comprise of two basic morphology operators such as dilation and erosion. The dilation is an operator that grows and thickens the objects in a binary image while the erosion operator erodes the foreground pixels. The basic effect of the erosion operator on a binary image is to erode away the boundaries of regions of foreground pixels (i.e. white pixels). The closing operator smoothes the boundaries of the image, while on the contrary the opening operator usually mixes the broken hybrids and omits the small details and fills the spaces of objects boundaries.

11.5 Concluding remarks

It can be said that every model that is developed has its own advantages and disadvantages. The more complex models may give better results based on accuracy and robustness, whereas the simpler ones need less processing power and can still give satisfying results. In this research, approximated median filtering was used in order to allow fast processing, but more robust algorithms should be developed to remove silhouette noise. The fall detector using different approaches was subjected to fall and non-fall experiments. Moreover, a variety of characteristics of different kinds of falls were included, which covered most of the types of falls suffered by elderly people, including frontal, sideways and backward falls.

Bibliography

[1] http://www.math_works.com Retrieved on 04.05.2010.

[2] S.E. Umbaugh. *Computer Vision and Image Processing*. National Instruments, 2nd edition, 2006.

[3] C. Rougier, A. St-Arnaud, J. Rousseau, and J. Meunier. Video surveillance for fall detection. *Video Surveillance, In Tech*, 2011.

[4] A. Sixsmith and N. Johnson. A smart sensor to detect the falls of the elderly. *Pervasive Computing, IEEE*, 3(2):42–47, 2004.

[5] C. Huang, E. Chen, and P. Chung. Fall detection using modular neural networks and back-projected optical flow. *Biomedical Engineering: Applications, Basis and Communications*, 19(6):415–424, 2007.

[6] http://www.learn_artificial_neural_networks.com Retrieved on 02.03.2010.

[7] http://www.doc.ic.ac.uk/surprise_96/journal/vol4/cs11/report.html Retrieved on 30.05.2010.

[8] R. Felder and M. Alwan. *Eldercare Technology for Clinical Practitioners*. Humana Press, 2008.

[9] Y. Zigel. A method fall detection of elderly people by using floor vibrations and sound-proof of concept on human mimicking doll falls. *Trans. on Biomedical Engineering, IEEE*, 56(9):2858–2867, 2009.

[10] J. Wang and S. Singh. Video analysis of human dynamics. *Real-Time Imaging*, 9(9):321–346, 2003.

[11] M. Alwan, G. Demiris, and Z. He. Technology for successful ageing. *Proc. of the 28th IEEE Embs Annual International Conference*, (5), 2006.

[12] Phac. Report on the falls of elder in Canada 2001 . *Proc. of the 28th IEEE Embs Annual International Conference*, (5). Report on the falls of elder in Canada 2001 `http://www.phacaspc.gc.ca/seniorsaines,Ottawa` Retrieved on 5.01.2010.

[13] U. Lindemann, A. Hock, and M. Stuber. Evaluation of a fall detector based on accelerometers: A pilot study. *Medical and Biological engineering and computing*, 43(5):548–551, 2005.

[14] T. Zhang, J. Wang, L. Xu, and P. Liu. Using wearable sensor and nmf algorithm to realize ambulatory fall detection. In *Advances in Natural Computation*, pages 488–491. Springer, 2006.

[15] E. B. Willson. *An Introduction to Scientific Research*. Dover Publications, 1991.

[16] H. Foroughi, S. Aski, and H. Pourreza. Intelligent video surveillance for monitoring fall detection of elderly in home environments. In *Computer and Information Technology, 2008. ICCIT 2008. 11th International Conference on*, pages 219–224. IEEE, 2008.

[17] J. Dai, X. Bai, Z. Yang, Z. Shen, and D. Xuan. Mobile phone-based pervasive fall detection. *Personal and ubiquitous computing*, 14(7):633–643, 2010.

[18] B. Seth. Background subtraction. August 2008.

[19] F. Sina, E. Michael, and M Peyman. Video-to-video dynamic super-resolution for grayscale and color sequences. *EURASIP Journal on Applied Signal Processing*, 2006:232–232, 2006.

[20] A. Bovik. *The essential guide to Image Processing.* Academic Press, 2009.

[21] A. Bovik. *Handbook of image and video processing.* Academic Press, 2010.

[22] R. Gonzales and R. Woods. *Digital Image Processing, 2-nd Edition.* Prentice Hall, 2002.

[23] `http://en.wikipedia.org/wiki/Mathematical_Morphology`Retrived on 24.04.2010.

[24] J. Munkres. *Image Analysis and Mathematical Morphology.* Prentice-Hall, 1999.

[25] J. Serra. *Image Analysis and Mathematical Morphology.* ISBN 0126372403, 1982.

[26] P. Soille. *Morphological Image Analysis; Principles and Applications.* 2nd edition, 2003.

[27] `http://www.dsp_guide.com/ch25/4.htm`.

[28] S. Cheung and C. Kamath. Robust techniques for background subtraction in urban traffic video. *Proc of the SPIE*, 5308:881–892, 2004.

[29] M. Piccardi. Background subtraction techniques: a review. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 4, pages 3099–3104. IEEE, 2004.

[30] R. Cucchiara, A. Prati, and R. Vexxani. A multi camera vision system for fall detection and alarm generation. *Expert Systems*, 24:334–345, 2007.

[31] S. Huwer and H. Niemann. Adaptive change detection for real-time surveillance applications. In *Visual Surveillance, 2000. Proceedings. Third IEEE International Workshop on*, pages 37–46. IEEE, 2000.

[32] N. J.B McFarlane and C. P. Schofield. Segmentation and tracking of piglets in images. *Machine Vision and Applications*, 8(3):187–193, 1995.

[33] C. Ridder, O. Munkelt, and H. Kirchner. Adaptive background estimation and ffreground detection using kalman-filtering. *Proc of International Conference on recent advances in mechatronics (ICRAM)*, 1995.

[34] G. Gordon, T. Darrell, M. Harville, and J. Woodfill. Background estimation and removal based on range and color. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2. IEEE, 1999.

[35] C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):747–757, 2000.

[36] M. Harville. A framework for high-level feedback to adaptive, per-pixel, mixture-of-gaussian background models. In *Computer Vision-ECCV 2002*, pages 543–560. Springer, 2002.

[37] W. Grimson, C. Stauffer, R. Romano, and L. Lee. Using adaptive tracking to classify and monitor activities in a site. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 22–29. IEEE, 1998.

[38] `http://www.u_tube.com/object_tracking` Retrived on 15.07.2010.

[39] N. Friedman and S. Russell. Image segmentation in video sequences: A probabilistic approach. In *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, pages 175–181. Morgan Kaufmann Publishers Inc., 1997.

[40] I. Cohen and G. Medioni. Detecting and tracking moving objects for video surveillance. *Computer Vision and Pattern recognition, IEEE Proc*, 1999.

[41] Q. Wan and Y. Wang. 'multiple moving objects tracking under complex scenes'. *The Sixth World congress on intelligent control and Automation (WCICA*, 2:9871–9875, 2006.

[42] F. Bunyak, I. Ersoy, and S. Subramanya. A multi-hypothesis approach for salient object tracking in visual surveillance. *International Conference on Image Processing, ICIP 2005, IEEE*, 2(II):446–449, 2005.

[43] A. Hampapur, L. Brown, J. Connell, A. Ekin, N. Haas, M. Lu, H. Merkl, and S. Pankanti. Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking. *Signal Processing Magazine, IEEE*, 22(2):38–51, 2005.

[44] G. Bradski and A. Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, 2008.

[45] D. Toth and T. Aach. Detection and recognition of moving objects using statistical motion detection and fourier descriptors. In *Image Analysis and*

Processing, 2003. Proceedings. 12th International Conference on, pages 430–435. IEEE, 2003.

[46] L. Shapiro. Stockman g: Computer vision, 2002.

[47] R. Luke, D. Anderson, J. Keller, and M. Skubic. Moving object segmentation from video using fused colour and texture features in indoor environments. *Journal of Real-Time Image Processing, Submitted for publication*, 2008.

[48] `http://www.definitions.net/definition/fall` Retrieved on 25.12.2009.

[49] `http://www.tunstall_group.com` Retrived on 05.01.2010.

[50] H. Meischke and E. Phelan. Falls and fall-related injures constitute a major public health. November.

[51] R. Tideiksaar. *Falling in old age*. Springer Publishing Company, 1996.

[52] K. Doughty, R. Lewis, and A. McIntosh. The design of a practical and reliable fall detector for community and institutional telecare. *J Telemed*, 6(suppl 1):S150–154, 2000.

[53] A. Ozcan, H. Donat, N. Gelecek, M. Ozdirenc, and D. Karadibak. The relationship between risk factors for falling and the quality of life in older adults. *BMC Public Health*, 5:90, 2005.

[54] R. Sattin, D. Huber, C. DeVito, J. Rodriguez, A. Ros, S. Bacchelli, J. Stevens, and R. waxweiler. The incidence of fall injury events among the elderly in a defined population. *Am. J. Epidemiol*, 131:1028–1037, 1990.

[55] A. Bourke and G. Lyons. A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. *Medical Engineering and Physics*, 30(1):84, 2006.

[56] T. Zhang, J. Wang, and L. Xu. Using wearable sensor and nmf algorithm to realize ambulatory fall detection. 30:488–491, 2006.

[57] K. Cameron, K. Hughes, and K. Doughty. Reducing fall incidence in community elders by telecare using predictive systems. *Proc. of the Engineering in Medicine and Biology society, 19th Annual International Conference of the IEEE*, 3(30):1036–1039, 1997.

[58] G. Williams, K. Doughty, K. Cameron, and D. Bradley. A smart fall and activity monitor for telecare applications. *Engineering in Medicine and Biology Society, Proceedings of the 20th Annual International Conference of the IEEE*, 3:1151–1154, 1998.

[59] H. Nait-Charif and S. J. McKenna. Activity summarisation and fall detection in a supportive home environment. *Proceedings of the Pattern recognition, ICPR 2004, 17th International Conference*, 4:323–326, 2004.

[60] A. Mihailidis, B. Carmichael, and J. Boger. The use of computer vision in an intelligent environment to support aging-in-place, safety, and independence in the home. *Information Technology in Biomedicine, IEEE Transactions on*, 8(3):238–247, 2004.

[61] B. Jansen and R. Deklerck. Home monitoring of elderly people with 3d camera technology. In *Proceedings of the first BENELUX biomedical engineering symposium, Brussels, Belgium*, 2006.

[62] B. Toreyin, Y. Dedeoglu, and A. Cetin. Hmm based falling person detection using both audio and video. *Proc IEEE International Workshop on Human-Computer Interaction, pp*, pages 211–220, 2005.

[63] D. Anderson, J. Keller, M. Skubic, X. Chen, and Z. He. Recognising falls from silhouettes. *International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 6388–6391, 2006.

[64] N. Thome and S. Miguet. A hhmm-based approach for robust fall detection. In *Control, Automation, Robotics and Vision, 2006. ICARCV'06. 9th International Conference on*, pages 1–8. IEEE, 2006.

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

[66] T. Y. Hsu, W. J. Hsieh, and Y. H. Liao. Human behaviour analysis using deformable triangulations. *IEEE 7th Wrkshop on Multimedia Signal Processing.*

[67] A. Williams, D. Ganesan, and A. Hanson. Aging in place: fall detection and localization in a distributed smart camera network. In *Proceedings of the 15th international conference on Multimedia*, pages 892–901. ACM, 2007.

[68] C. Lin and Z. Ling. Automatic fall incident detection in compressed video for intelligent homecare. *Computer Communications and Network, ICCCN*, pages 1172–1177, 2007.

[69] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau. Monocular 3d head tracking to detect falls of elderly people. In *Proceedings of the 28th IEEE EMBS Annial.* International Conference, 2006.

[70] N. Noury, P. Rumeau, A. Bourke, G. ÓLaighin, and J. Lundy. A proposal for the classification and evaluation of fall detectors. *Irbm*, 29(6):340–349, 2008.

[71] M. Kangas, A. Konttila, P. Lindgren, I. Winblad, and T. Jämsä. Comparison of low-complexity fall detection algorithms for body attached accelerometers. *Gait & posture*, 28(2):285–291, 2008.

[72] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G Celler. Implementation of a real-time human movement classifier using a tri-axial accelerometer for ambulatory monitoring. *Information Technology in Biomedicine, IEEE Transactions on*, 10(1):156–167, 2006.

[73] A. Bourke and G. Lyons. A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. *Medical Engineering and Physics*, 30(1):84, 2008.

[74] M. Nyan, F. E. Tay, and E. Murugasu. A wearable system for pre-impact fall detection. *Journal of biomechanics*, 41(16):3475–3481, 2008.

[75] M. Alwan, P. J. Rajendran, S. Kell, D. Mack, S. Dalal, M. Wolfe, and R. Felder. A smart and passive floor-vibration based fall detector for elderly. In *Information and Communication Technologies, 2006. ICTTA'06. 2nd*, volume 1, pages 1003–1007. IEEE, 2006.

[76] S. Londei, J. Rousseau, F. Ducharme, A. St-Arnaud, J. Meunier, and F. Giroux. An intelligent video monitoring system for fall detection at gome@ perceptions of elderly people. *Journal of Telemedicine and Telecare*, 15(8):383–390, 2009.

[77] J. Willems, G. Debard, B. Bonroy, B. Vanrumste, and T. Goedeme. How to detect human fall in video? an overview. *In Positioning and Context-Awareness International Conference (POCA) 2009, Antwerp, Belgium*, 2009.

[78] J. Boyle and M. karanunanithi. Simulated fall detection via accelerometers. *Int IEEE-EMBS, Vancouver*, 2008.

[79] W. Chia-Chi. Development of fall detection system for the elderly residents. *The 2nd International Conference on Bioinformatics and Biomedical Engineering, ICBBE*, 2008.

[80] K. Alan and K. Bourke. Fall detection through vertical velocity thresholding using a tr-axial accelerometer characterised using an optical motion-capture system. *30th Annual International IEEE EMBS Conference*, 2008.

[81] M. Narayanan. Falls management; detection and prevention, using a waist-mounted triaxial accelerometer. *Int. Conf. of the IEEE-EMBS Cite International, Lyon*, 2007.

[82] N. Noury. A smart sensor for the remote follow up of activity and fall detection of the elderly. *Int Conf. IEEE-EMBS 'Micro technologies in Medicine and Biology', Madison-USA*, 2002.

[83] M. Popescu, Y. Li, M. Skubic, and M. Rantz. An acoustic fall detector system that uses sound height information to reduce the false alarm rate. *Engineering in Medicine and Biiology Society, EMBS, the Annual International Conference of the IEEE*, 30:4628–4631, 2008.

[84] S. Ng, A. Fakih, A. Fourney, P. Poupart, and J. Zelek.

[85] T. Lee and A. Mihailidis. An intelligent emergency response system: preliminary development and testing of automated fall detection. *Journal of telemedicine and telecare*, 11(4):194–198, 2005.

[86] D. Anderson, R. H. Luke, J. M. Keller, M. Skubic, M. Rantz, and M. Aud. Linguistic summarization of video for fall detection using voxel person and fuzzy logic. *Computer Vision and Image Understanding*, 113(1):80–89, 2009.

[87] E. Auvinet, L. Reveret, A. St-Arnaud, J. Rousseau, and J. Meunier. Fall detection using multiple cameras. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 2554–2557. IEEE, 2008.

[88] N. Thome, S. Miguet, and S. Ambellouis. A real-time, multiview fall detection system: A lhmm-based approach. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(11):1522–1532, 2008.

[89] D. Aarno and D. Kragic. Motion intention recognition in robot assisted applications. *Robotics and Autonomous Systems*, 56(8):692–705, 2008.

[90] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau. Fall detection from human shape and motion history using video surveillance. In *21st International Conference on Advanced Information Networking and Applications Workshops*, 2007.

[91] L. Young-Sook and L. HoonJae. Multiple object tracking for fall detection in real-time surveillance system'. *In proceedings of the 11th International conference on Advanced Communication Technology (ICACT) 2009, Phoenix Park, Republic of Korea. IEEE Press*, pages 2308–2312, 2009.

[92] G. Debard, P. Karsmakers, M. Deschodt, E. Vlaeyen, J. Van den Bergh, E. Dejaeger, K. Milisen, T. Goedemé, T. Tuytelaars, and B. Vanrumste. Camera based fall detection using multiple features validated with real life video. In *Workshop Proceedings of the 7th International Conference on Intelligent Environments*, volume 10, pages 441–450, 2011.

[93] A. William, D. Ganesan, and A. Hanson. Ageing in place: Fall detection and localisation in distributed smart camera network. *Proceedings of the 15th*

International Conference on Multimedia, burg, Germany, pages 892–901, Augs 2007.

[94] V. Vishwakarma, C. Mandal, and S. Sural. Automatic detection of human fall in video. *Lecture Notes in Computer Science, Pattern Recognition and Machine Intelligence,*, 4815:616–623, 2007.

[95] M. Russell and R. Moore. Explicit modelling of state occupancy in hidden markov models for automatic speech recognition. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'85.*, volume 10, pages 5–8. IEEE, 1985.

[96] B. Juang and L. R. Rabiner. Hidden markov models for speech recognition. *Technometrics*, 33(3):251–272, 1991.

[97] I. Nabney. *Netlab-Algorithms for pattern recognition*. Springer, 2001.

[98] Z. Fu, T. delbruxk, P. Lichtsteiner, and Culurciello E. An address-event fall detector for assited living applications. *IEEE Transactions on Biomedical Circuits and Systems*, 2(2):88–96, 2008.

[99] D. Anderson, R. H. Luke, J. M. Keller, and M. Skubic. Extension of a soft-computing framework for activity analysis from linguistic summarisations of video. In *Fuzzy Systems, 2008. FUZZ-IEEE 2008.(IEEE World Congress on Computational Intelligence). IEEE International Conference on*, pages 1404–1410. IEEE, 2008.

[100] M. Luštrek and B. Kaluža. Fall detection and activity recognition with machine learning. *Informatica*, 33(2):197–204, 2009.

[101] J. Berclaz, F. Fleuret, and P. Fua. Multi-camera tracking and atypical motion detection with behavioral maps. In *Computer Vision–ECCV 2008*, pages 112–125. Springer, 2008.

[102] T. Zhang, J. Wang, L. Xu, and P. Liu. Fall detection by wearable sensor and one-class svm algorithm. In *Intelligent Computing in Signal Processing and Pattern Recognition*, pages 858–863. Springer, 2006.

[103] G. Sukthankar and K. Sycara. A cost minimization approach to human behaviour recognition. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 1067–1074. ACM, 2005.

[104] F. Marquis-Faulkes, S. McKenna, A. Newell, and P Gregor. Gathering the requirements for a fall monitor using drama and video with older people. *Technology and Disability*, 17(4):227–236, 2005.

[105] C. Juang and C. Chang. Human body posture classification by a neural fuzzy network and home care system application. *IEEE Transactions on systems, man and cybernetics*, 37(6):984–994, 2007.

[106] `http://www.statsoft.com/text_book/neural_networks` Retrieved on 26.05.2010.

[107] M. Rafiq, G. Bugmann, and D. Easterbrook. Neural network design for engineering applications. *Computers & Structures*, 79(17):1541–1552, 2001.

[108] L. V. Fausett. *Fundamentals of neural networks: architectures, algorithms, and applications*. Prentice-Hall Englewood Cliffs, 1994.

[109] `http://en.wikipedia.org/wiki/Artificial_Neural_Network` Retrieved on 28.05.2010.

[110] M. Sordo. Introduction to neural networks in healthcare. *Open Clinical Document*, 2002.

[111] http://en.wikipedia.org/wiki/Sigmoid_function Retrieved on 30.06.2010.

[112] S. N. Haykin. A comprehensive foundation. *Neural Networks*, 2, 2004.

[113] K. Mehrotra, C. K. Mohan, and S. Ranka. Elements of artificial neural networks. a bradford book, 1997.

[114] J. Paola and R. Schowengerdt. A review and analysis of backpropagation neural networks for classification of remotely-sensed multi-spectral imagery. *International Journal of remote sensing*, 16(16):3033–3058, 1995.

[115] http://www.learn_artificial_neural_networks.com/back_propagation.html Retrieved on 24.06.2010.

[116] S. Kotsiantis, I. Zaharakis, and P. Pintelas. Supervised machine learning: A review of classification techniques. *Frontiers in Artificial Intelligence and Applications*, 160:3, 2007.

[117] http://en.wikipedia.org/wiki/Supervised_learning Retrieved on 15.07.2010.

[118] V. Vapnik. *The nature of statistical learning theory*. springer, 1999.

[119] http://tph.tuwien.ac.at/~oemer/doc/neurogen/node5.html Retrieved on 22.05.2010.

[120] D. Beasley, D. R . Bull, and R. R. Martin. An overview of genetic algorithms: Part 2, research topics. *University computing*, 15(4):170–181, 1993.

[121] L. Chambers. Genetic algorithms, 2001.

[122] K. Deb. *Optimization for engineering design: Algorithms and examples.* PHI Learning Pvt. Ltd., 2004.

[123] M. Pelikan, D. E. Goldberg, and F. G. Lobo. A survey of optimization by building and using probabilistic models. *Computational optimization and applications*, 21(1):5–20, 2002.

[124] P. Charbonneau. An introduction to genetic algorithms for numerical optimization. 2002.

[125] J. H. Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control and artificial intelligence.* MIT press, 1992.

[126] `http://en.wikipedia.org/wiki/Greyscale` Retrieved on 25.07.2010.

[127] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. volume 77, pages 257–286. IEEE, 1989.

[128] N. Noury, A. Fleury, P. Rumeau, A. Bourke, G. Laighin, V. Rialle, and J. Lundy. Fall detection-principles and methods. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 1663–1666. IEEE, 2007.

[129] M. Lan, A. Nahapetian, A. Vahdatpour, L. Au, W. Kaiser, and M. Sarrafzadeh. Smartfall: an automatic fall detection system based on subsequence matching for the smartcane. In *Proceedings of the Fourth International Conference on Body Area Networks*, page 8. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.

[130] T. Degen, H. Jaeckel, M. Rufer, and S. Wyss. Speedy: a fall detector in a wrist watch. In *Proceedings of the 7th IEEE International Symposium on Wearable Computers*, page 184. IEEE Computer Society, 2003.

[131] J. Hwang, J. Kang, Y. Jang, and H. Kim. Development of novel algorithm and real-time monitoring ambulatory system using bluetooth module for fall detection in the elderly. In *Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE*, volume 1, pages 2204–2207. IEEE, 2004.

[132] W. L. Buntine. Operations for learning with graphical models. *Journal of Artificial Intelligence Research*, 2:159–225, 1994.

[133] M. O. Nuria, R. Barbara, and P. P. Alex. A bayesian computer vision system for modeling human interactions. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 22(8):831–843, 2000.

[134] K. P. Murphy. *Dynamic bayesian networks: representation, inference and learning*. PhD thesis, University of California, 2002.