



UNIVERSITY OF
PORTSMOUTH

*Context Fusion for Recognising Activities of Daily Living from
Indoor Video Data*

By

Mohamad Zuhair Saeed Al-Wattar

The thesis is submitted in partial fulfilment of the requirements for the award
of the degree of Doctor of Philosophy of the University of Portsmouth

April 2020

Declaration

Whilst registered as a candidate for the above degree, I have not been registered for any other research award. The results and conclusions embodied in this thesis are the work of the named candidate and have not been submitted for any other academic award.

Mohamad Zuhair Saeed Al-Wattar

Acknowledgement

I would like to express my sincere gratitude to my supervisor Dr Rinat Khusainov for his continuous support and guidance throughout my study. His feedback, motivation and positivity helped me through all the obstacles I faced during my research. Dr Khusainov was always keen to help in my research development, I learned countless lessons from him, and I am delighted and proud to have had him as my supervisor.

Besides my supervisor, I would like to thank the academic staff at the University of Portsmouth, represented by the School of Energy and Electronic Engineering, for being always supportive and for providing me with an excellent environment and wise advice during my PhD study.

Thanks to the Higher Committee for Education Development / Iraqi Prime Minister Office for their valuable support. Without their precious help, it would not have been possible for me to conduct this research. Also, thanks to the Iraqi Ministry of Communications for allowing me to pursue further studies.

I would also like to thank my family, and friends for their encouragement. Your wise counsel, sympathetic ear, and continuous support kept me going throughout my years of study. I cannot thank you enough for all the support and guidance you provided me.

Special thanks to my friend Dr Harith Kharrufa for his encouragement, help, and support during the period of my study.

Last but not least, I would like to thank Professor David Hogg and Professor Anthony G Cohn from the school of computing at the University of Leeds, for giving me the opportunity to stay and work at their facility, and for their precious advice, support, and help in my PhD placement.

Dedication

To my Father and Mother...

You are the reason I kept going. Thank you for your great support.

You have been my inspiration.

To my brothers and sister...

Thank you for your support along the way.

Abstract

Human activity recognition using images and videos is an area that is undergoing intensive research in the field of computer vision. During the last few years, the topic has attracted many researchers due to its benefit in a number of applications, including surveillance systems and assisted living. There is a significant need for an automated system that can help the elderly and enable them to live independently. As the elderly population is increasing, in the near future, there will not be enough care workers to provide them with the necessary help.

The available activity recognition methods target low-level activities, posture activities, and a few high-level activities. The techniques used in identifying high-level activities depend mainly on extracting the low-level features and using these to identify the activities. Although some of these techniques have achieved high performance in identifying certain high-level activities, there are both essential and instrumental activities of daily living that have not yet been recognised. In addition, these methods are trained in identifying staged activities that are recorded in unrealistic conditions and locations, and do not use high-level features to identify activities.

The advancement in computer systems, parallel computing, and the development of deep learning frameworks enable the creation of an automated system that can identify human activities and take the necessary actions based on the recognised activity. This facilitates the use of complicated machine learning algorithms that help in creating a system to support the elderly and people with special needs in living independently and performing their daily activities. This can be achieved by using different algorithms for identifying features and activities based on a sequence of images.

In this thesis, three major contributions are presented. The first is the creation of a method for identifying high-level activities of daily living and falls using high-level features. These high-level features are spatial (i.e., location of the detected person), temporal, the detected person's posture, and their orientation. The method uses different models of machine learning algorithms including the Convolution Pose Machine, classifiers, the Long Short-Term Memory model, and the Hidden Markov model, for extracting high-level features from

sequences of images, and uses these extracted high-level features to identify high-level activities

The second contribution is the creation of a dataset called PortAD that addresses and solves the major issues that other existing datasets suffer from. This dataset is then used to evaluate the effectiveness of the proposed method to identify high-level activities. PortAD overcomes many of the limitations of the available datasets, including missing activities, non-realistic locations, non-practical locations for cameras, and an inadequate number of cameras used. In this work, 14 activities of daily living and instrumental activities of daily living are recorded using multiple cameras located in the top corners of the rooms.

The third major contribution is an evaluation of the effectiveness of the selected four high-level features in identifying activities of daily living. Multiple activity recognition models are proposed to identify activities of daily living, including the fixed and adaptive time threshold model, the Hidden Markov Model, and the Long Short-Term Memory.

The first finding of the study is that, combined, the selected four high-level features achieved better results compared to using one, two and three features. This is due to the fact that individual features may have problems that can be overcome when used in combination with other features.

The proposed approaches were successfully tested and evaluated via practical experiments using two datasets, the PortAD dataset, and a fall dataset that is used to identify sudden activity. High-level activities were identified using the Forward algorithm, the Forward-Backward algorithm, and the Long Short-Term Memory model; these achieved accuracy rates of 92.88%, 93.01%, and 93.32%, respectively, when tested on PortAD, and accuracy rates of 84.9%, 87.5%, and 87.7%, respectively, when tested on the fall dataset.

Table of Contents

Declaration	i
Acknowledgement	ii
Dedication.....	iii
Abstract	iv
Chapter 1 Introduction.....	1
1.1 Research Area	1
1.2 Motivation.....	3
1.3 Aims and Objectives	4
1.4 Contributions	5
1.5 Thesis outline	5
Chapter 2 Literature Review.....	7
2.1 Activities of Daily Living.....	7
2.2 Activity Monitoring Systems	8
2.3 Data Models	9
2.3.1 Hidden Markov Model.....	10
2.3.2 Neural Networks	12
2.3.3 Convolution Neural Network.....	13
2.3.4 Recurrent Neural Network	16
2.4 System Platforms.....	20
2.4.1 Wearable Sensors.....	21
2.4.2 Environmental Sensors	24
2.5 Data Processing.....	32
2.5.1 Hidden Markov Model.....	32
2.5.2 Neural Networks	39

2.5.3 Hybrid Approach	45
2.5.4 Pose Estimators.....	47
2.6 System Evaluation.....	52
2.6.1 Image Datasets.....	53
2.6.2 Video Datasets	55
2.6.3 Metrics.....	66
Chapter 3 Research Scope	69
Chapter 4 Identifying High-Level Activities	75
4.1 Selecting Features for Activity Identification.....	75
4.1.1 Nontechnical Justifications	75
4.1.2 Technical Justifications	77
4.2 Spatial and Temporal Features	79
4.2.1 Identifying Location.....	79
4.2.2 Applying Time Threshold	81
4.2.3 Applying Hidden Markov Models.....	85
4.3 Posture and Orientation Features	91
4.3.1 Convolutional Pose Machine	91
4.3.2 Posture and Orientation Classification.....	94
4.4 Combining Spatio-Temporal and Pose-Based Features.....	100
4.4.1 Hidden Markov Models	100
4.4.2 Long Short-Term Memory	105
4.5 Complete Design.....	108
Chapter 5 – Evaluation of High-Level Activity Identification.....	113
5.1 Dataset Requirements.....	113
5.2 Experimental Environment	117
5.2.1 Port-Eco House.....	117

5.2.2 Home Office	119
5.2.3 Cameras	119
5.2.4 Network and Servers	120
5.2.5 Privacy and Security	122
5.3 Experimental Scenarios	123
5.3.1 Home Office Activities	124
5.3.2 Common Domestic Activities	127
5.4 Comparison with Existing Datasets	130
Chapter 6 Effectiveness of Features and Fusion Methods	133
6.1 Conducted Experiments	133
6.2 Data Preparation	134
6.3 Algorithm Parameters Selection	138
6.4 Performance Metrics	140
6.5 Hardware and Software	140
6.6 Results	141
6.7 Analysis	149
6.8 Comparison of Feature Performance and Limitations	156
Chapter 7 Conclusion	158
7.1 Discussion of Contributions	158
7.2 Future Work	161
References	165
Appendix A	191

List of Figures



Figure 2. 1: A block diagram for the activity monitoring system.....	9
Figure 2. 2: The states for HMM – A is the hidden states; T is the transition probability; E is the emission probability; and L is the observable states.	11
Figure 2. 3: Neural network, where the Xs are the inputs, Y the output, Ws are the weights, and Hs are the hidden layer neurons.....	12
Figure 2. 4: NN vs DNN.	13
Figure 2. 5: 2D feature map creation.	14
Figure 2. 6: Pooling with a 2*2 window and a stride of two – the colours represent the max pooling window, and the value represents the highest number as it is max pooling.....	15
Figure 2. 7: Sample for RNN showing the time steps, three inputs and three outputs, and how the nodes connected.....	16
Figure 2. 8: A sample LSTM cell and its application to processing sequential inputs.	18
Figure 2. 9: Flowchart for the activity recognition system (Wang, Huang & Tan, 2007).	33
Figure 2. 10: The merging of three different HMM, $P(O \lambda)$ is computed based on the three likelihood probabilities $P(O \lambda_g)$, $P(O \lambda_a)$ and $P(O \lambda_k)$. (Liu et al., 2014).	34
Figure 2. 11: Jalal et al. (2015) human activity recognition system (Jalal, Kamal & Kim, 2015).	35
Figure 2. 12: Activity recognition from Kinect (Gaglio, Re & Morana, 2015).....	35
Figure 2. 13: Block diagram for the activity recognition system from a depth camera with HMM (Uddin et al., 2016).....	36
Figure 2. 14: The relationship between the depth of the network and the network accuracy (Goodfellow et al., 2016).....	40
Figure 2. 15: The rank pooling hierarchical activity recognition system (Fernando et al., 2016).	44
Figure 2. 16: The articulated human detection and human pose estimation (Yang & Ramanan, 2013).....	49
Figure 2. 17: The complete framework for the CPM (Wei et al., 2016).....	50
Figure 2. 18: Example from the LSP dataset.....	53
Figure 2. 19: Sample from the HPE dataset with upper body annotation.....	54

Figure 2. 20: Sample from the FLIC dataset.	54
Figure 2. 21: Sample from the KTH dataset.....	55
Figure 2. 22: Sample of the Weizmann dataset.....	56
Figure 2. 23: The TUM Kitchen dataset.....	57
Figure 2. 24: Six action classes from UCF101.	58
Figure 2. 25: Sample from the MPII Cooking Activities dataset.	58
Figure 2. 26: The Epic Kitchen dataset.	59
Figure 2. 27: The CAD-60 dataset.	60
Figure 2. 28: The CAD-120 dataset.	61
Figure 2. 29: The 16 activities in the MSR DailyActivity dataset.	62
Figure 2. 30: Multiview 3D Event dataset.	63
Figure 2. 31: Sample of the Sphere-H130 dataset.....	63
Figure 2. 32: Sample from the multiple cameras fall dataset.	64
Figure 2. 33: Sample from the fall detection dataset.	65
Figure 2. 34: UR Fall Detection dataset.....	65
Figure 4. 1: Shows the hand labelling for the objects in the locations.....	80
Figure 4. 2: Represents the location detection using the BS-KNN on the left side, and the CPM on the right side, the arrow represents the selected point for the BS-KNN and the CPM.....	81
Figure 4. 3 The system design based on spatial and temporal features.....	82
Figure 4. 4: The pseudocode for the fixed time threshold algorithm	83
Figure 4. 5 The activity based on the fixed threshold output	84
Figure 4. 6: The pseudocode for the adaptive time threshold algorithm.....	84
Figure 4. 7 Block diagram for HMM with spatial and temporal features	86
Figure 4. 8: The pseudocode for the Forward algorithm	88
Figure 4. 9: The pseudocode for the Forward-Backward algorithm.....	91
Figure 4. 10 The identifies body parts by the CPM (Wei et al., 2016)	92
Figure 4. 11 The detection for a trained CPM	93
Figure 4. 12 The right image shows a partially occluded person, and the image on the left shows how the CPM manage to identify the human body parts (with some errors)	93
Figure 4. 13: The pseudocode for the normalisation process.....	95

Figure 4. 14: The completed design for the normalisation with the classification method for identifying the posture and orientation algorithm.....	96
Figure 4. 15: Activities that can be easily identified by the Pairwise method	98
Figure 4. 16: Posture and orientation classification using pairwise distances.....	99
Figure 4. 17: LSTM network design.....	106
Figure 4. 18: The input and the output for the LSTM unit where ht is the current hidden state, $ht - 1$ is the previous hidden state, Ct is the cell state, and $Ct - 1$ is the previous cell state. Xt is the input vector.	107
Figure 4. 19 HMM model for the four features. $A(t)$ denotes activity at time t and is the hidden state; $POL(t)$ represents the combination of posture, orientation, and location as the observed states.	109
Figure 5. 1 Port-Eco House	117
Figure 5. 2 Ethernet ports and power socket in a corner of a room.	118
Figure 5. 3 The control room in PEH	118
Figure 5. 4 Home office	119
Figure 5. 5 The camera network for the PEH	121
Figure 5. 6 Samples of the data in the kitchen and home office.....	123
Figure 5. 7 The detected person working on the computer	125
Figure 5. 8 Camera views for the PortAD dataset	130
Figure 6. 1: LSTM window size for training and testing, when the window is equal to five (P=posture, O=orientation, L=location)	137
Figure 6. 2: Activity identification performance on the PortAD dataset (A-Min= Adaptive threshold Minimum, A-Max= Adaptive threshold Maximum, A-Avg= Adaptive threshold Average, HMM-F= Hidden Markov Model the Forward algorithm, HMM-FB =Hidden Markov Model the Forward-Backward algorithm). The main bars show the average values, and the error bars show the minimum and maximum values for each metric.	143
Figure 6. 3: The fall detection performance on the UR fall dataset using the proposed methods and features (HMM-F= Hidden Markov Model the Forward algorithm, HMM-FB =Hidden Markov Model the Forward-Backward algorithm).....	145

Figure 6. 4: The accuracy of detecting posture and orientation on the PortAD dataset. The main bars show the average value, and the error bars show the minimum and maximum values for each case.	147
Figure 6. 5: The accuracy of detecting posture and orientation on the UR fall dataset	148
Figure 6. 6: The reason for improving the performance when adding the orientation. The labelled activity for the left picture is using the sink, and the labelled activity for the right picture is walking. The two activities share the same location and the same posture.....	151
Figure 6. 7: The reason for improving the performance when adding the posture: the labelled activity for the left picture is preparing a meal (it is part of ‘preparing a meal’ activity), and the labelled activity for the right picture is eating. Two activities share the same location and same orientation.....	152
Figure 6. 8: Errors in location detection using the BS-KNN: the black arrows show the correct location (manually labelled), while the red arrows show the location detected with BS-KNN	153
Figure 6. 10: Some of the problems in posture and orientation detection	155
Figure 7. 1 Object detection using CNN (YOLO9000).....	164

List of Tables

Table 2. 1 The three different types of ADL (Lawton, 1990; Kempen & Suurmeijer, 1990; Kempen, Myers & Powell, 1995)	8
Table 2. 2: The confusion matrix explaining the TP, FP, FN, and TN.	66
Table 3. 1 Potential research problems for a video-based assisted living system.....	72
Table 4. 1: Likely problems in using location only for activity recognition (AL: Actual Location, DL: Detected Location, AA: Actual Activity, Ln: Location, An: Activity, Tn: Time).....	78
Table 4. 2 An example of the coordinated values for the fourteen points from the CPM output	94
Table 4. 3 the normalised values for the fourteen coordinates from the CPM output for Table 4. 2.....	95
Table 5. 1 Specification for the two selected camera models (Foscam)	120
Table 5. 2 Home office activities, with the day of the week, time of day, activity name and duration	127
Table 5. 3 Scenarios for three days in the PEH kitchen, for three meals.....	129
Table 5. 4 Summary of popular activity recognition datasets and PortAD.....	131
Table 6. 1: The activity identification performance on the PortAD dataset ('Min', 'Max', and 'Avg' correspond to the minimum, maximum, and average performance across different cameras; Fixed 5 f/sec = The fixed time threshold, when the threshold is 5 frames/sec; A-Min= Adaptive minimum threshold; A-Max= Adaptive maximum threshold; A-Avg= Adaptive average threshold; HMM-F= HMM with the Forward algorithm; HMM-FB =HMM with the Forward-Backward algorithm)  Represent the highest values for each feature combination.  Represent the highest values across all feature combinations	142
Table 6. 2: The performance values on data from the selected camera in the UR fall dataset using the proposed methods and features (HMM-F= Hidden Markov Model the Forward	



algorithm, HMM-FB =Hidden Markov Model the Forward-Backward algorithm).  Represent the highest values for each feature combination.  Represent the highest values across all feature combinations..... 144



Table 6. 3: The accuracy of detecting posture and orientation on the PortAD dataset ('Min', 'Max', and 'Avg' correspond to the minimum, maximum, and average performance across different cameras).  Represent the highest values for each feature combination.  Represent the highest values across all feature combinations. 146



Table 6. 4: The accuracy of detecting posture and orientation on the UR fall dataset.  Represent the highest values for each feature combination.  Represent the highest values across all feature combinations..... 148

Table 6. 5: Pearson correlation coefficient values for the three conditional probability assumptions used to estimate emission probabilities in HMM when using all four high-level features. 148

Table 6. 6: A comparison between the selected features 157

Chapter 1 Introduction

1.1 Research Area

Human Activity Recognition (HAR) is one of the most active topics in the field of computer vision. It has drawn much attention and has been used in many applications, such as video annotation and retrieval, human–computer interaction, and video surveillance (Aggarwal & Ryoo, 2011). The aim of HAR is to identify human activities from a number of observed features; these features can be, for instance, spatial, temporal, colour, or audio.

Assisted Living (AL) is one of the areas that require HAR, as it aims to support the elderly and people with special needs in living independently and performing their daily activities. The primary goal of AL is to help the people above to live independently and improve the safety of living in their environment (Dohr, Modre-Opsrian, Drobics, Hayn & Schreier, 2010). In addition, AL can help support the health and wellbeing of the elderly while they perform their daily work and activities from their home, through medication reminders and management, fall identification and prevention, and contacting emergency services when needed (Rashidi & Mihailidis, 2013).

Single-person activities in an indoor environment can be divided into three main types: posture activities, low-level activities, and high-level activities. Posture activities are those that are constructed from gesture movements, and include standing, sitting, walking, jumping, and punching. A gesture movement is a body part movement by the person, such as raising a hand, lifting a leg, or stretching an arm (Aggarwal & Ryoo, 2011). Low-level activities are those that are constructed by a body part movement with an object, or occur in a specific place, such as cutting and using a computer mouse. Posture activities and low-level activities can construct high-level activities, such as cooking or working on a computer. Multiple people can perform group activities that relate to a single object, such as carrying an object or multiple objects; one example of a group activity would be a band playing music.

Different types of sensors are in use currently and they are integrated into daily activity gadgets and garments, such as wearable sensors (i.e. wristwatches, headphones) and environmental sensors (i.e. cameras and ultrasonic sensors). Many researchers have studied wearable sensors in recent years. The majority of the wearable sensors can connect to cloud

services, which makes it easier for researchers to collect data to train and improve the activity recognition system.

Currently, many companies in health care sector are focusing on developing assisted living technologies using different approaches. Some of them use attached hardware and wearable devices, such as Withings smartwatch, the Apple Watch, and Fitbit activity monitors. These types of sensor achieve a reasonable level of accuracy in measuring some daily activities, including standing, sitting, and falling. They can act as a heart rate monitor, step counter, or calorie calculator. Some companies, such as Philips, have focused on providing help via telephone and mobile applications.

Philips created its telecare system to provide help to the elderly. The Philips Telecare solution (Care Sage) is offered on a monthly subscription basis, and requires an action from the patient when a problem occurs, either to call or to press a button within the application. In some cases, this represents a significant issue. Some companies use wearable technology to identify physical activities and detect falls. The primary limitation of existing wearable technologies is that they provide a limited amount of information, which minimises the benefits of their use. In addition, they are noticeable and easy to lose; some people find them unattractive and an invasion of their private space. In addition, to be effective they have to be worn all the time, and there is a concern that the individual may forget to wear the device. Furthermore, wearable devices require batteries, which must be charged, creating further concern.

The human brain processes visual information faster than other information, as 65% of people are visual learners (Bradford, 2004; Sauseng & Klimesch 2008; Anderson & Hinton, 2014). Hence, researchers have focused on images and videos to identify activities, utilising a variety of image sensors (cameras). Image sensors have many features compared to other types of sensors. For standard coloured RGB (red, green, and blue) images, there are the dimensions (resolutions) of the images and RGB values. Some companies, such as Ocuvera, have used optical sensors to prevent falls, with impressive results; however, this prevents falls only when the algorithm predicts that patients will leave their bed, by sending a message to the responsible person or carer, to alert them that the patient requires help.

Multiple models have been used to identify activities from different types of features, such as probabilistic approaches (the Hidden Markov Model), and fixed approaches (location-

based approaches). Although human activity recognition systems have been extensively researched in the last 30 years, there is still a long way to go before a complete system is achieved, due to the many challenges faced, such as occlusion, lighting, environment, scaling, time, and location.

There are many benefits to identifying human activities, from the most straightforward application of creating an automated system that turns devices on/off when a person enters or leaves a room, to understanding human interactions and interpersonal relations, identifying the physical or psychological status of a monitored person, and helping a monitored person to live independently.

1.2 Motivation

With the increase in life expectancy (Roser, Ortiz-Ospina & Ritchie, 2013), there is a clear need for more systems that help people in need and the elderly. As the older population increases, there is a shortage of care workers to provide them with the necessary help and support. According to the national population projection in the UK (Nash, 2016), there will be more than 16 million people aged 65 and over by 2035, and the number of carers is not sufficient to provide them with the support that they need. Therefore, technology-based solutions are essential. Researchers and companies are working on providing these solutions using different types of sensors and methods. Specifically, researchers are working on identifying the activities of daily living in order to help the elderly and people in need to live comfortably and longer, by preventing and reducing the impact of sudden activities and facilitating a more natural way of life in their home environment. Utilising technology can also help reduce the need for private care homes.

The ability to identify a more comprehensive range of activities of daily living (ADL), and instrumental activities of daily living (IADL) is necessary to achieve a reliable and efficient telecare system. The available systems identify only a limited number of ADL and IADL, or focus on low-level activities which are not essential components of ADL and IADL. A high-performance system that can identify the critical ADL and IADL in addition to sudden activities and that can work in real-time is currently lacking.

Researchers typically test proposed systems using datasets of recorded staged activities, not real-life activities; this can affect the system's performance when tested in real-life. The

available datasets cover many activities, but there are some activities yet to be covered. Furthermore, most of the available datasets were recorded in labs. Therefore, there is a need for a dataset recorded in a real location, working in real-time and tested in a real-life scenario. In addition, the majority of the existing datasets have been recorded by a single camera located at a body-height or head-height location. That means they cover a smaller area, and the view can be easily occluded. A dataset recorded using multiple cameras located in the top corners of the location will overcome these issues.

A cheap, expandable, easy to install a system with minor modifications is also lacking. The available systems are either challenging to install and maintain, such as the raised floor system to identify gait, using nonvisual sensors, or using non-IP cameras that affect the expandability of the system. Consequently, this motivated the researcher to study and assess the available technologies and create an affordable system that can identify high-level activities and sudden activities in a real environment. The aim of this research is also to provide recommendations for future developments based on the experimental results and findings obtained from this work.

1.3 Aims and Objectives

Many researchers are showing interest in the area of activity recognition and its applications. Most of this work focuses on identifying posture activities, low-level activities and some high-level activities using low-level features.

The main aim is to create a method for identifying high-level activities using high-level features. There are many studies on activity recognition, but as discussed, there is still a need for further investigations. The contribution of this thesis will be in combining the four high-level features, namely: location, time, posture, and orientation of the detected person.

As the study takes an evaluation approach to high-level activity identification, a dataset was created that covers the essential ADL and IADL. This dataset was recorded and labelled based on the aforementioned high-level features in order to assess the performance of the proposed system.

The ultimate purpose is to study the effectiveness of the selected four high-level features on the performance of the system in identifying activities, to understand the impact of these four features on the performance of identifying the daily activities.

1.4 Contributions

The main contributions of the thesis are as follows:

- The creation of a novel method of combining spatial and temporal information with the position and orientation of the detected person to identify activity; i.e., using four high-level features to identify high-level activities, by using different methods and combining different models of machine learning algorithms to predict high-level activities and sudden activities. The methods used to identify these activities are:
 - The fixed time threshold
 - The adaptive time threshold
 - The Hidden Markov Model
 - The Forward algorithm
 - The Forward–Backward algorithm
 - The Long Short-Term Memory
- The creation of a real-life dataset, the Portsmouth Activity Dataset (PortAD), used for method evaluation and labelling the dataset based on the four selected high-level features.
- Evaluating the effectiveness of the four high-level features for identification of activity and sudden activity, by addressing the usefulness of those features in identifying high-level activities and sudden activities.

1.5 Thesis outline

This PhD thesis is composed of a total of seven chapters, structured as follows:

Chapter 2 presents a review of the related literature, and the methods used to identify activities. It also provides an overview of the hardware most commonly used in activity recognition systems, and the algorithms used in the field. Finally, it introduces the datasets and the metrics that are used to evaluate previous researchers' work.

Chapter 3 presents the possible working areas in ADL and IADL, and the literature gaps that this thesis is intended to fill.

Chapter 4 provides an overview of the four high-level features (location, time, posture, and orientation) used in this study, and justifies their selection. Then, it presents the proposed methods for identifying ADL and IADL based on two features, starting with fixed and adaptive threshold methods, then HMM and HMM with a classifier. The complete method utilising all four features is then presented, and the way in which the features are extracted and combined using multiple models of machine learning algorithms is explained.

Chapter 5 explains the approaches used to evaluate the proposed methods. First, the motivation for creating PortAD is explained, followed by a description of the working areas and locations in which the videos were recorded. Then, the network and servers used in this work are presented, and the activities that PortAD covers are shown. Finally, a comparison between PortAD and the popular activity datasets is presented.

Chapter 6 examines the effectiveness of the selected high-level features in identifying activities. First, the experiments are described, and then the dataset preparations. Then, the parameters of the selected algorithms are presented. Later in the chapter, the metrics used to evaluate the work and the hardware and software employed are described. The results achieved using the four features and the proposed methods are presented, followed by justification and analysis of the results. Finally, a comparison between the selected high-level features and used methods and their impact on the performance is presented.

Chapter 7 concludes the thesis and provides recommendations for future development in the area.

Chapter 2 Literature Review

According to the UK Office of National Statistics, in the next 20 years there will be more than 16 million people aged 65 and above in the UK alone. This number will continue increasing, and there will not be enough care workers to look after the people who need help (Knickman & Snell, 2002). Industry and academic researchers are working on various solutions and techniques to try to find an answer to this critical problem. Researchers have applied various methods using different types of sensors to solve this problem, for example, smart camera systems, wearable sensors, and smart homes.

This chapter reviews the previous works related to this research area. First, the key concepts of activities of daily living and instrumental activities of daily living are explained, as well as sudden activities. Then, the equipment and sensors used by previous researchers in this field are presented, followed by the technologies used in activity recognition and similar applications, beginning with the Hidden Markov Model, and then neural network. Later in the chapter is a discussion of how researchers have combined the Hidden Markov Model and neural networks in their work. At the end of the chapter, the datasets and the metrics used to test and evaluate the existing methods are presented.

2.1 Activities of Daily Living

Indoor and outdoor Activities of Daily Living are series of physical events that are performed by people on a daily basis. These activities help people to maintain their independence in their living environment and the community (Lawton, 1990; Mlinac & Feng, 2016). Researchers have separated Activities of Daily Living into two categories: Activities of Daily Living, or basic ADL; and Instrumental Activities of Daily Living (IADL) (Lawton, 1990; Kempen & Suurmeijer, 1990; Kempen, Myers & Powell, 1995).

Basic ADL refers to the activities that are the minimum requirement for an individual to perform the most straightforward actions; the IADL are more complex activities that are fundamental to allowing individuals to live independently in their home (see Table 2. 1). For example, eating is an ADL, while taking pills is an IADL. The reason for this distinction is that ADL are driven by natural bodily needs, and people need to eat in order to sustain themselves. Taking pills, on the other hand, usually requires a fixed timing, and some medicines come with

restrictions, such as needing to be administered before or after meals. Therefore, IADL require greater cognitive awareness.

Sudden activities do not fall under the ADL or IADL categories. However, in this work, they are added to the indoor and outdoor ADL. Sudden activities can affect anyone in any location, at any time, for instance due to accidents or medical reasons, such as gait and balance disorders, sleep disorders, and obesity (Salzman, 2010), or external reasons, such as pushing or pulling a person.

Table 2. 1 The three different types of ADL (Lawton, 1990; Kempen & Suurmeijer, 1990; Kempen, Myers & Powell, 1995)

Activities of Daily Living		
Basic (ADL)	Instrumental (IADL)	Sudden
Standing	Housekeeping (cleaning and maintaining)	Falling - Sit to fall - Stand to fall - Lie to fall
Sitting	Financial management (managing money, working at home)	
Lying (bed/sofa)	Moving within the community	
Hand washing	Preparing meals/cooking	
Bathing	Shopping	
Showering	Taking medicine	
Hygiene (personal hygiene, toilet hygiene)	Using the telephone	
Grooming	Housekeeping	
Dressing	Care of others	
Eating/self-feeding	Care of pets	
Walking/movement/locomotion	Communication management	
Standing to sit	Religious observances	
Sitting to standing	Safety procedures and emergency responses	
Sitting to lying		
Lying to sitting		
Lying to standing		

2.2 Activity Monitoring Systems

An activity monitoring system is a complete platform that consists of hardware and software components that are used to detect a person, track the detected person, identify the activity

for the detected person, and provide actions when needed based on the detected activity. A typical monitoring system is shown in Figure 2. 1.

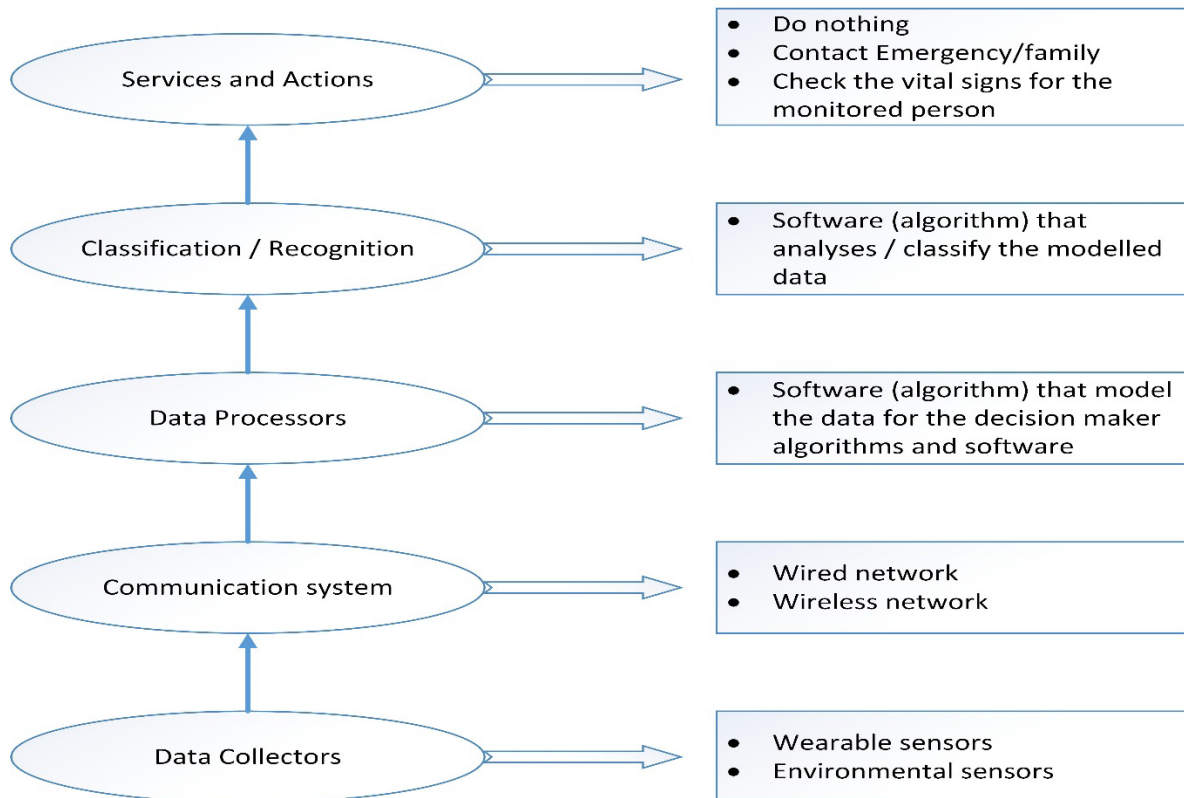


Figure 2. 1: A block diagram for the activity monitoring system.

The first step for any activity monitoring system is to collect the data, which can be done by humans or by sensors. The collected data needs to be carried securely to the data processors via a wired or a wireless network. The data processors will then convert the collected data into a format that the model (classification/recognition) can use to identify the activity. Based on the results that the model produces, the services and actions will provide the most suitable action to the monitored person. Typical services and actions could be contacting a health worker, calling or live streaming the detected person, or checking the person’s vital signs (remotely or locally).

2.3 Data Models

The Hidden Markov Model (HMM) has been widely used by many researchers. There are many significant uses of HMM, such as speech recognition (Rabiner, 1989), handwriting

identification (Hu, Brown & Turin, 1996; Gales & Young, 2008), face recognition and detection (Nefian & Hayes, 1998), and internet traffic identification (Munther, Othman, Alsaadi & Anbar, 2016).

As HMM has achieved promising results, researchers have used it with single and multiple features. For instance, Gaikwad (2012) used HMM to identify activities from a single feature, while Kijak et al. (2003) combined more than one feature, in their case audio and video data. Another approach is to use multiple HMMs together, such as Coupled HMM (Brand, Oliver & Pentland, 1997), and Layered HMM (Kabir, Hoque, Thapa & Yang, 2016).

Neural networks are new popular approaches being used by researchers (Goodfellow, Bengio & Courville, 2016). Neural networks such as the Feedforward Neural Network and the Convolution Neural Network have been used by many researchers due to their ability to extract features and perform the tasks that are trained for with high performance.

2.3.1 Hidden Markov Model

The Hidden Markov Model (HMM) has been widely used in solving classification and pattern recognition problems in speech, voice, and activity recognition (Rabiner, 1989; Brailovskiy & Herman, 2014). HMM is a statistical model that represents the time series data and the hidden state from the observed state. HMM also represents the probability distribution over sequences of observation for each state. There are sets of probabilities, called the transition probabilities and the emission probabilities (Yamato, Ohya & Ishii, 1992; Gilks, Richardson & Spiegelhalter, 1995; Ghahramani, 2001; Hallinan, 2012). HMM is a sequential process where, at each step, the hidden states change according to the transition function, and the observations are generated according to the observation function. The transition probability is the probability of moving from one state to a different state, shown in Equation 2. 1 and Figure 2. 2. The emission (observation) probabilities are the probabilities of the different outputs given the current state – see Equation 2. 2 and Figure 2. 2.

HMM is made up of a transition probability matrix $P(A_t|A_{t-1})$ and an emission probability matrix $P(L_t|A_t)$, and consists of two types of states, the hidden states and the observable states. The hidden states are the states of the system described by the Markov process, such as the activities; the observable states are those that are visible, such as location.

An $HMM = (T, E, \pi)$, is shown in Figure 2. 2, where:

- π represents a probability distribution over the initial HMM states: $\pi(a) = P(A_1 = a)$.
- T represents the transition probability function for HMM, and is calculated as shown below:

$$T(A_t, A_{t-1}) = P(A_t | A_{t-1}, A_{t-2}, A_{t-3}, \dots, A_1) = P(A_t | A_{t-1}), \quad 2.1$$

where A_t represents the hidden state at time t and A_{t-1} is the previous hidden state. This expression is based on the Markovian assumption used in HMM, stating that the next state in HMM depends only on the current state and not on how we arrived to this state (i.e. all the previous states).

- E represents the emission (observation) probability function as shown below:

$$E(L_t, A_t) = P(L_t | A_t), \quad 2.2$$

where L_t is the observed state at time t and A_t is the hidden state at time t .

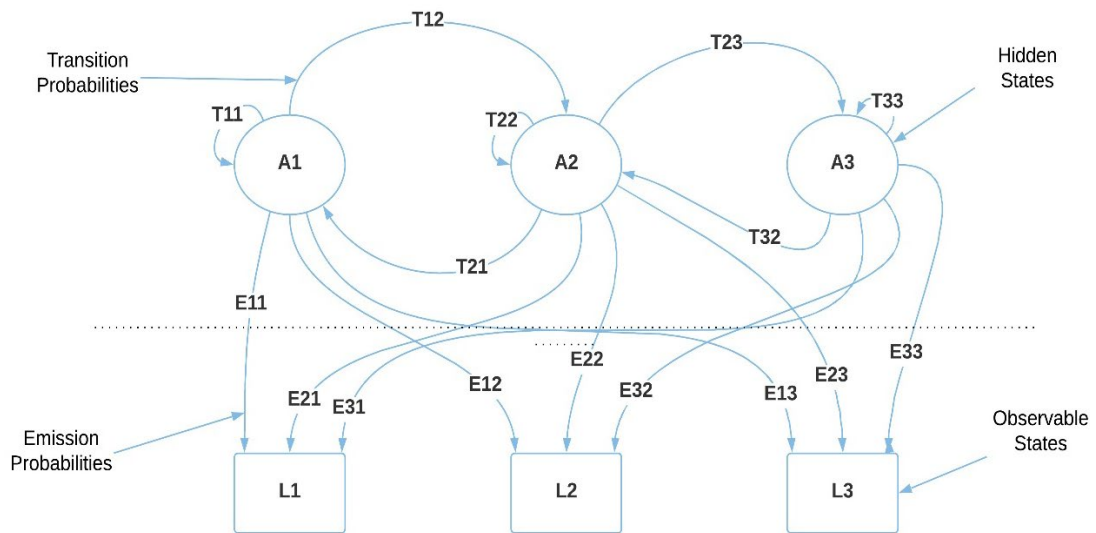


Figure 2. 2: The states for HMM – A is the hidden states; T is the transition probability; E is the emission probability; and L is the observable states.

To create an HMM model, all the parameters for HMM need to be addressed: the hidden states, the observed states, the initial states, the emission probabilities, and the transition probabilities, as shown in Figure 2. 2.

2.3.2 Neural Network

A neural network (NN) is as a set of mathematical equations and powerful machine learning algorithms that can solve many problems, such as classification, recognition, feature learning, and regression, based on the training data that is fed into the algorithm. The reason for calling these neural networks is that the algorithms are attempting to identify the relationships between the data in a way that mimics the human brain (Priddy & Keller, 2005), as shown in Figure 2. 3.

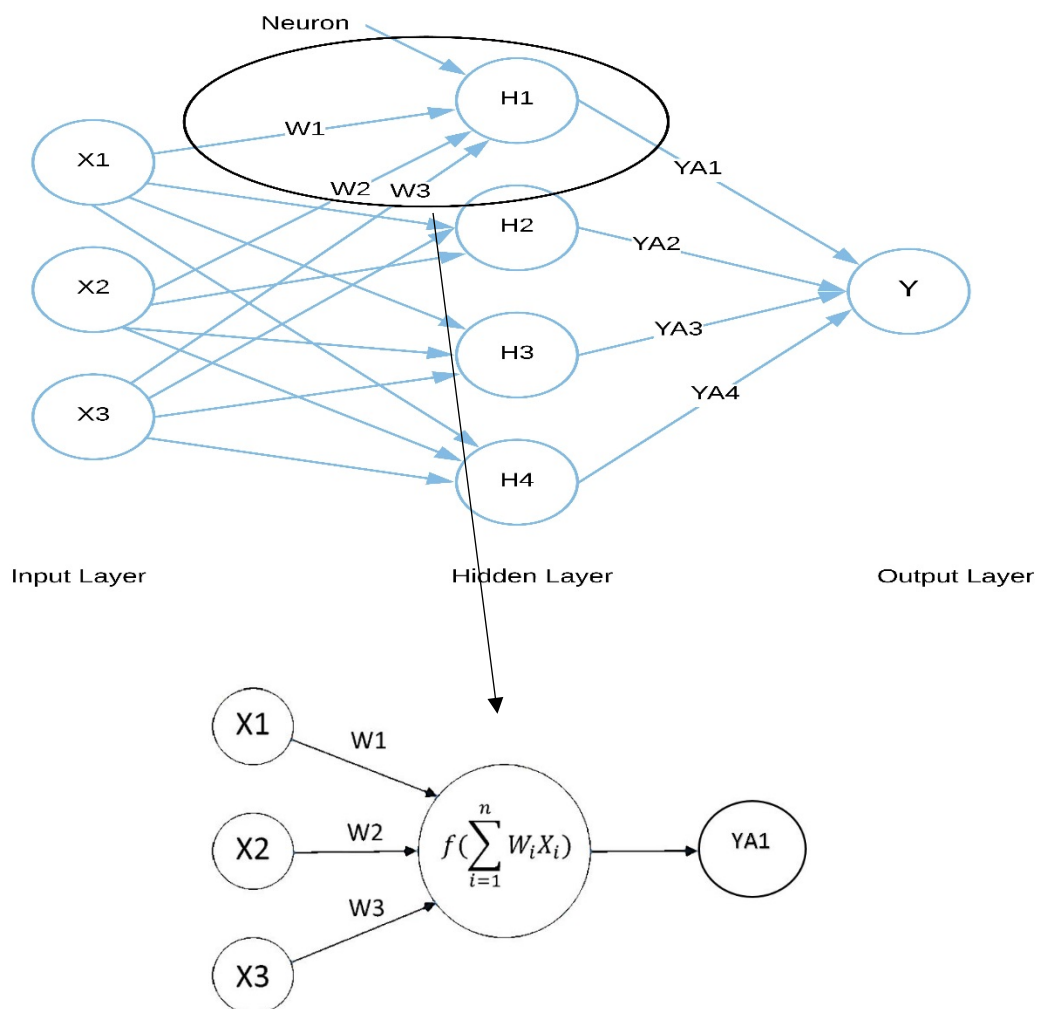


Figure 2. 3: Neural network, where the Xs are the inputs, Y the output, Ws are the weights, and Hs are the hidden layer neurons.

Each neural network has an input layer, hidden layer/layers, and an output layer. The hidden layer has neurons, and each neuron has weights, biases, and an activation function. At each neuron in the hidden layers, the weight with the input is added to the bias, and an activation function is then used to produce the output, as shown in Figure 2. 3 and Equation 2. 3.

$$Y = f \left(\sum X_i W_i \right) + b$$

2.3

- Y is the output
- f is the transfer (activation function)
- X is the inputs
- W is the weights
- b is the bias

A deep neural network is a subtype of neural network with more hidden layers. In general, when the number of hidden layers is more than two, this means that the neural network has more depth, and can be called a deep neural network. A deep neural network is shown in Figure 2.4.

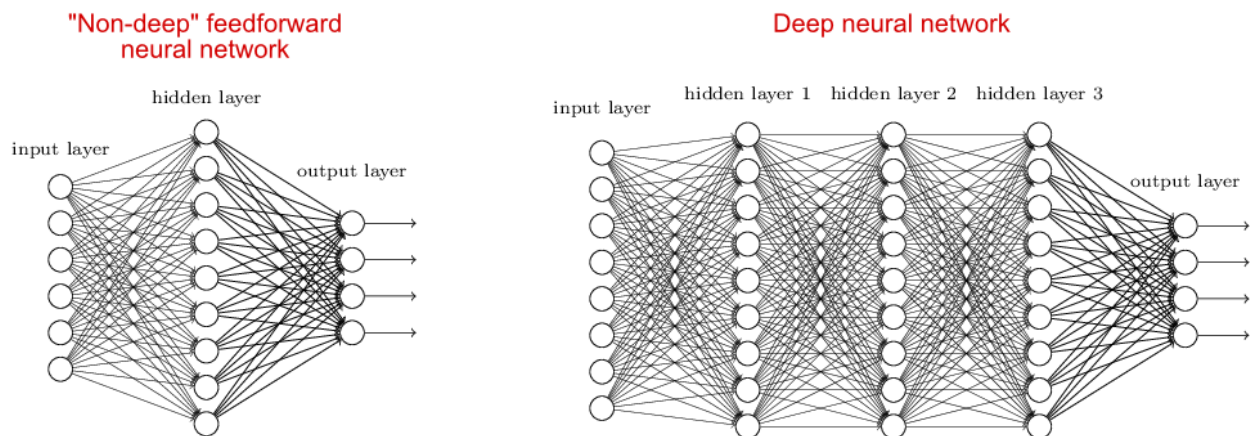


Figure 2.4: NN vs DNN.

2.3.3 Convolution Neural Network

The Convolution Neural Network (CNN or ConvNet) is a type of neural network that is generally used for image identification, classification, object detecting, or segmentation. The objective of image segmentation is to make the image easier to analyse by changing the representation of the image to make it more meaningful (Shapiro & Stockman, 2001; Barghout & Lee, 2004).

CNN is similar to Feedforward NN, and consists of neurons that have learnable weights and biases. The input layer for the CNN is usually an image, and the output is a class. The hidden layers consist of convolutional layers, activation function, pooling layers, fully connected layers, and backpropagation (LeCun & Bengio, 1995; Sermanet, Chintala & LeCun, 2012).

2.3.3.1 Convolutional layer

The convolutional layer is the main part of the ConvNet. Each image contains an enormous number of features; therefore, using Feedforward NN will be extremely slow and thus consume a tremendous amount of time to process a single image. For instance, if the image size is $720 \times 1280 \times 3$ (width, height, and depth), this means it will have 2.7 million features, meaning 2.7 million weights for each neuron if the fully connected Feedforward NN is used, which will take an extremely long time to process. Therefore, the convolutional layers are commonly used as a method to select the features (LeCun & Bengio, 1995).

The convolutional layer is a feature extractor method that uses a kernel (filter) to select the useful features from the input image. The convolutional layer will apply the kernel that will select the most relevant features, which will allow the network to go deeper with a lower computational process. An example of a 2D filter is shown in Figure 2. 5. In addition, in the convolutional layer, when the padding is disabled – such as ignoring the zero paddings – the number of features produced in the feature map will be lower, which will help in reducing the processing time.

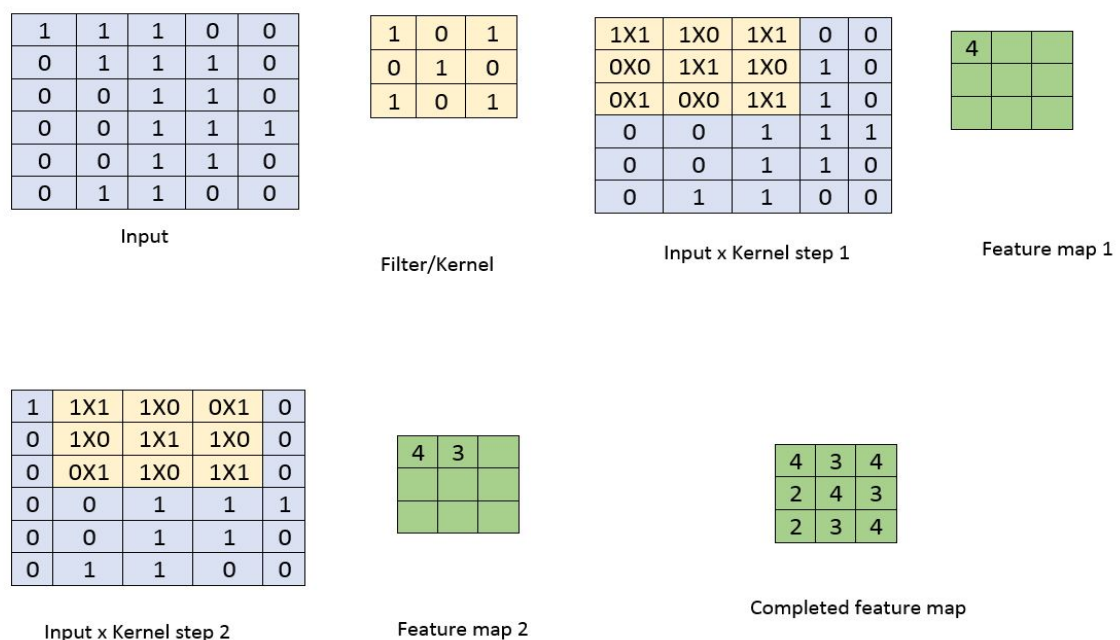


Figure 2. 5: 2D feature map creation.

Multiple filters can be used to produce the feature map; the number of the kernels will determine the number of feature maps, which will determine the depth. If five kernels are used, this means the number of features maps will be five, and the depth will also be five.

2.3.3.2 Activation function

The activation function, such as Rectified Linear Unit *ReLU*, is used after each convolutional layer, to normalise all the values in the feature map and prevent linearity. *ReLU* will return '0' for any negative values, and the value itself for positive values. The number of values in the feature map will not be changed. *ReLU* will convert all the negative values to zero, as shown in the equation below:

$$ReLU = \max(0, input) \quad 2.4$$

There are other non-linear activation functions, such as *tanh* and *sigmoid*, but researchers have found that *ReLU* outperforms them in many models (Krizhevsky, Sutskever & Hinton, 2012; Jia et al., 2014; Maturana & Scherer, 2015).

2.3.3.3 Pooling layers

The pooling layers are down-sampling layers that will reduce the total number of features. The pooling layer will reduce the dimensionality of the kernel map, as shown in Figure 2. 6. Reducing the number of parameters will control the overfitting and reduce the processing time. The pooling layers are down-sampling layers with no weights and do not require any training.

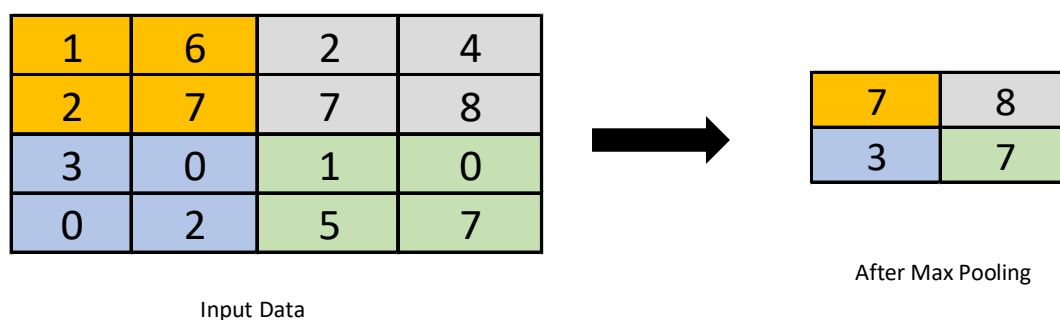


Figure 2. 6: Pooling with a 2*2 window and a stride of two – the colours represent the max pooling window, and the value represents the highest number as it is max pooling.

2.3.3.4 Fully Connected Layer

The fully connected layer is a multi-layer perceptron that uses an activation function in the output layer. 'Fully connected' means that every neuron in the layer is connected to every

neuron in the next layer; more details were provided earlier in section 2.3.2. The fully connected layer is presented in Figure 2. 3 and Figure 2. 4.

2.3.4 Recurrent Neural Network

Recurrent Neural Networks (RNN) are neural networks which take the output from the previous state (the hidden states) in addition to the input in order to produce the output, as shown in Figure 2. 7. Compared to NNs, RNNs can take a sequence of inputs and return either a sequence of outputs or a single output.

RNNs are used in multiple applications, such as natural language processing and speech recognition (Graves, Mohamed & Hinton, 2013; Miao, Gowayed & Metze, 2015) because of its feedback connection, which can learn sequential data (Schmidhuber, 2015), as the computation at each step considers the context of the previous steps in the form of the hidden states.

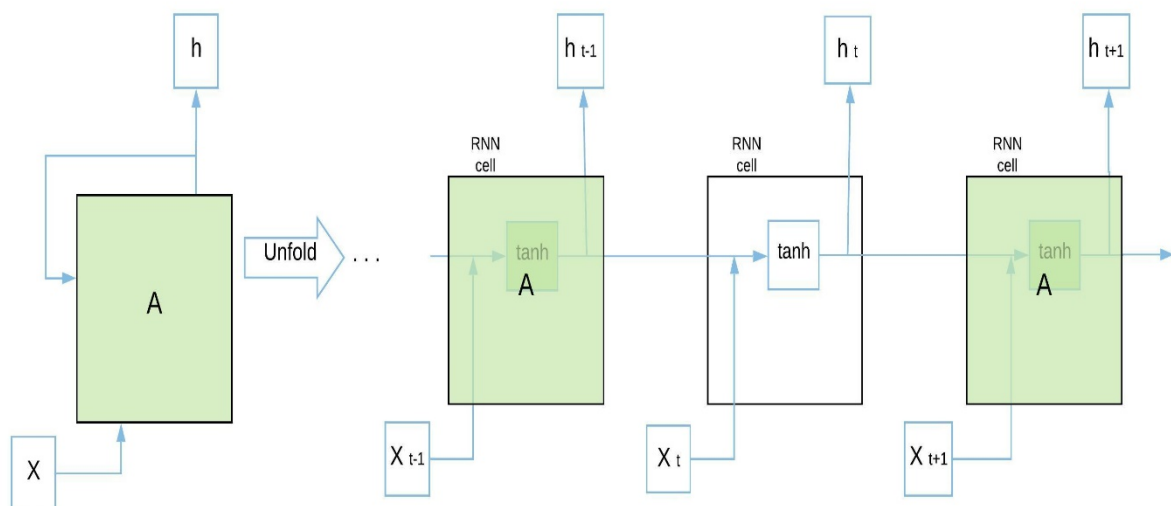


Figure 2. 7: Sample for RNN showing the time steps, three inputs and three outputs, and how the nodes connected.

RNN is similar to Feedforward NN. Feedforward NN learns through backpropagation and RNN learns using backpropagation through time. The formulas for RNN are slightly different from those the Feedforward NN because it has two inputs (the hidden states and the input vectors):

$$h_t = f(h_{t-1}, X_t) \tag{2.5}$$

$$h_t = \tanh(W_h \times h_{t-1} + W_X \times X_t) ,$$

where h_t is the new hidden state, X_t is the current input value, h_{t-1} is the previous hidden state, \tanh is the activation function, W_h is the weight of the hidden state, W_x is the weight of the input X_t .

The output values are calculated as follows:

$$Y_t = W_Y \times h_t , \quad 2.6$$

where Y_t is the current output, and W_Y is the weight of the output Y_t .

The cost function is used to measure the errors between the predicted and the true output values during the training of the network. Based on the cost function, the model can be tuned. Gradient descent algorithm is used to calculate the optimum values for the weight values in the network.

Although RNNs are used in multiple applications, their main limitation is the vanishing gradient (Bengio, Simard & Frasconi, 1994). During the backpropagation, the gradient values diminish when going backwards in the network; this leads to a slower learning speed for the neurons in the earlier layers compared to the neurons in the later layers. This can affect the overall performance of the network. Therefore, researchers have adopted approaches that overcome this issue, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) (Hochreiter & Schmidhuber, 1997; Cho, Van Merriënboer, Bahdanau & Bengio, 2014; Chung, Gulcehre, Cho & Bengio, 2014).

The first, LSTM (Hochreiter & Schmidhuber, 1997) is designed to overcome the vanishing gradient problem by remembering the information for a long time (Olah, 2015), by using gates inside the LSTM cell itself. LSTM cells have three gates – an input gate, an output gate, and a forget gate – which helps to regulate the flow of information from one cell to the next. This helps in remembering the sequence of input information. An LSTM cell and its application to processing sequential inputs are shown in Figure 2. 8.

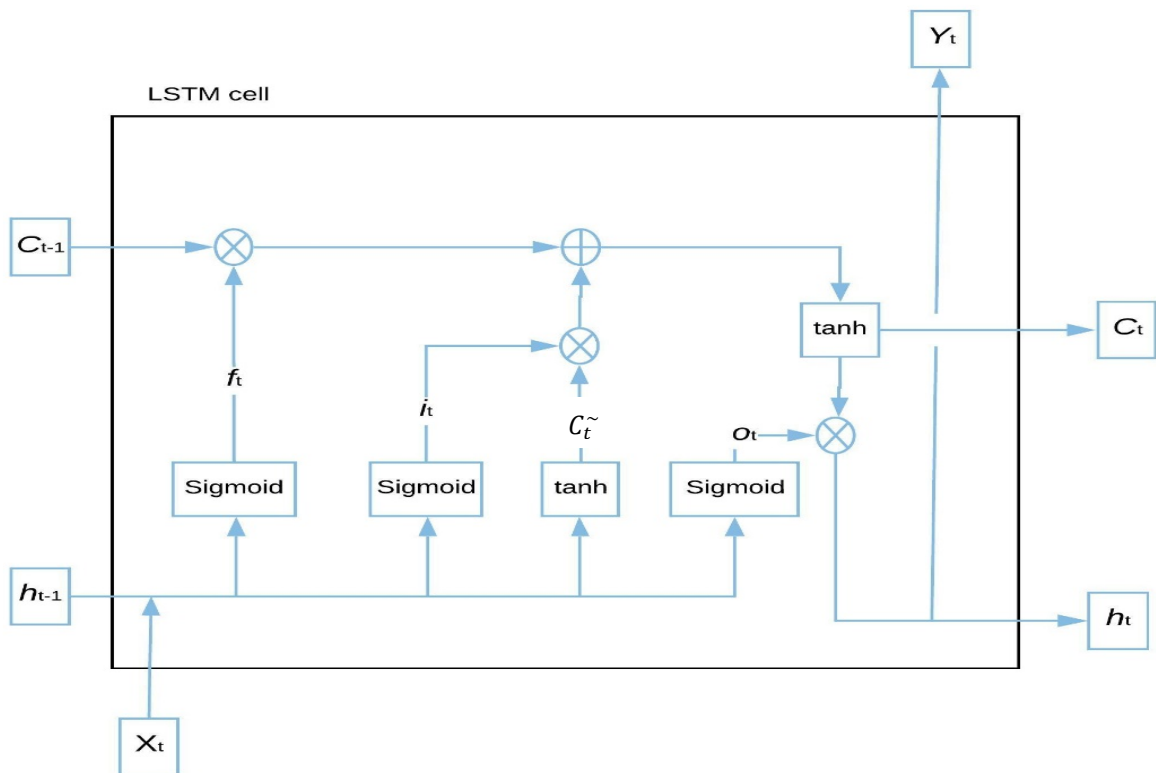
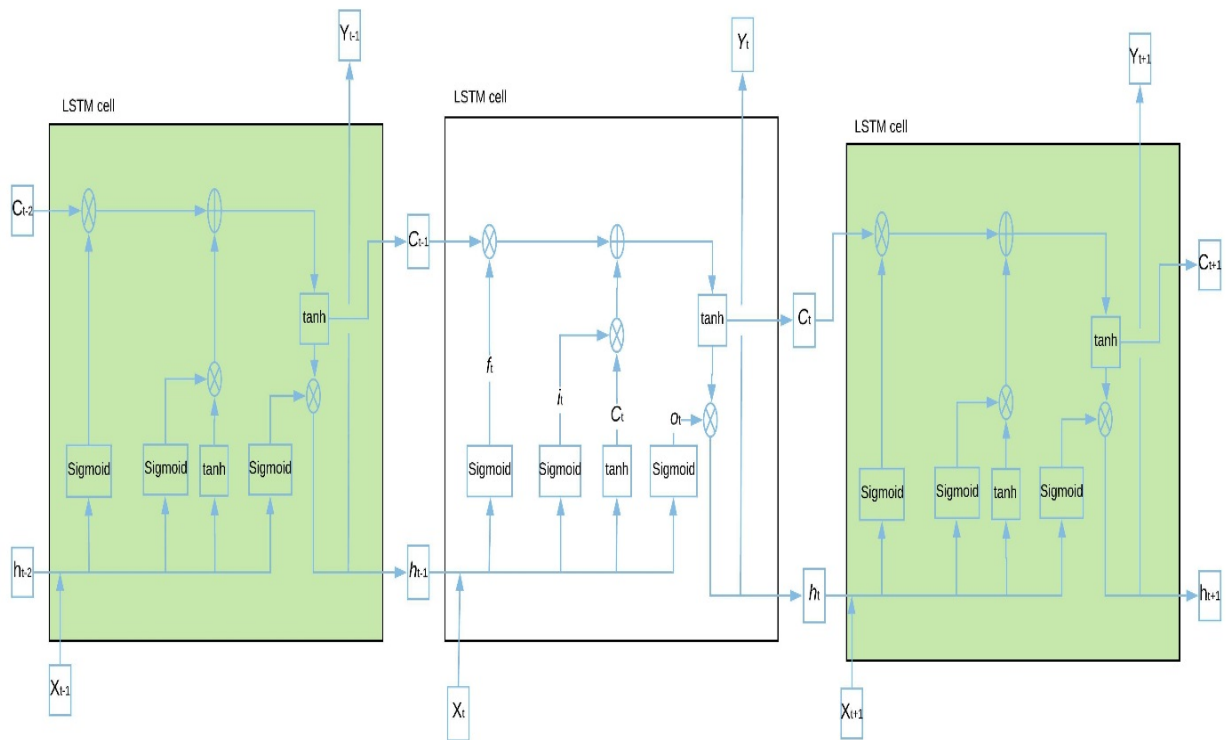


Figure 2. 8: A sample LSTM cell and its application to processing sequential inputs.

The forget gate decides which information to keep and which is not relevant to the model, using the sigmoid function, which will convert the outcomes to values between “0” and “1”. Those values that are close to 0 will be ignored, and the values that are closer to 1 will be used. The formula for the forget gate is presented below (Hochreiter & Schmidhuber, 1997; Olah, 2015):

$$f_t = \sigma(W_f(h_{t-1}, X_t)) + b_f, \quad 2.7$$

where f_t is the forget gate output, σ is the sigmoid activation function, W_f is the weight of the forget gate, and b_f is the bias value for the forget gate.

The input gate will take the sigmoid for the combination of previous output (the previous hidden h_{t-1} state) and the current input X_t . The sigmoid will select the values that are more relevant to the model (closer or equal) to 1 and ignore the less important values. The input gate equation is shown in the following formula (Hochreiter & Schmidhuber, 1997; Olah, 2015):

$$i_t = \sigma(W_i(h_{t-1}, X_t)) + b_i, \quad 2.8$$

where i_t is the input gate output, W_i is the weight of the input gate, and b_i is the bias for the input gate.

At the same time, the combination of the previous hidden state h_{t-1} and the current input X_t , shown in Figure 2. 8, will pass through the \tanh activation function. The \tanh activation function will rescale the values to between “-1” and “1”, to help regulate the network. This is shown in the following equation for the candidate value.

$$C_t^{\sim} = \tanh(W_c(h_{t-1}, X_t)) + b_c, \quad 2.9$$

where C_t^{\sim} is the candidate value, \tanh is the activation function, and b_c is the bias for the candidate.

Then calculate the cell state shown in Figure 2. 8. The forget gate values f_t will be multiplied by the previous cell state values C_{t-1} and then added to the input state i_t multiplied by the candidate value C_t^{\sim} . The cell state formula is shown below:

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \quad 2.10$$

The output gate is the last one to calculate, and will decide the next hidden state h_t . To calculate the output gate, the previous hidden state and the current input will go into a sigmoid function, as shown in the below formula:

$$o_t = \sigma(W_o(h_{t-1}, X_t)) + b_o, \quad 2.11$$

where o_t is the output gate, W_o the weight of the output gate, and b_o is the bias for output gate.

Finally, to calculate the hidden state h_t or the output Y_t , the current cell state is passed to the *tanh* activation function and then multiplied by the output gate output o_t as shown in the below equation:

$$h_t = o_t \times \tanh(C_t) \quad 2.12$$

Then, the hidden state will be used to predict the activity, using a *softmax* function:

$$Y_t = f(h_t) \quad 2.13$$

As LSTM is similar to Feedforward NN, the depth of the network may improve the performance, but it will increase the optimisation complexity. Different algorithms can be used for weight initialisation and optimisation, such as the Xavier initialisation method (Glorot initialiser) (Glorot & Bengio, 2010), He initialisation (He, Zhang, Ren & Sun, 2015), the adaptive subgradient method (Adagrad) (Duchi, Hazan & Singer, 2011), and adaptive moment estimation (ADAM) (Kingma & Ba, 2014).

2.4 System Platforms

Different hardware and software have been used to identify activities and sudden activities. This section focuses on the hardware component of activity recognition systems. Several types of sensors have been used to collect data in activity recognition systems, in particular optical (image) sensors; orientation sensors that can sense any movement (accelerometer and gyroscope sensors); bolometer sensors to read the thermal heat that emanates from the detected object; proximity sensors; heart rate sensors and IR sensors to sense and detect movement; and vital signs sensors for action recognition.

For the purposes of this study, the sensors used in such systems are split into two main categories: wearable sensors and environmental sensors. The wearable sensors are all those that are in direct contact with the monitored person's body, while the environmental sensors are those that are not in direct physical contact with the monitored person's body.

2.4.1 Wearable Sensors

Motion sensors have been employed to identify human activities based on spatial information (Guesgen, 2015). Sensors have been used to recognise activities based on temporal data. In particular the time of the day, the day of the week, and seasons have been used to improve activity recognition system performance (Aztiria, Augusto, Izaguirre & Cook, 2009).

Wearable sensors include wristbands, systems attached to a person's belt, an over shoulder pouch, and systems worn on the head or the chest. Such systems can measure the vital signs of the monitored person, and can help to prevent some diseases by giving a warning when a reading is higher or lower than the set thresholds. Some of these sensor systems can provide a heart rate reading, the frequency of leg movements, body temperature, end-tidal carbon dioxide, blood oxygen saturation, and blood pressure (Park & Jayaraman, 2003).

Moy, Mentzer and Reilly (2003) created a system using accelerometers that monitors patients with chronic obstructive pulmonary disease (COPD), in what is considered one of the first research studies in this area. The system takes readings and keeps track of the monitored patients while they are performing their daily tasks. This system demonstrates the potential clinical applicability of using accelerometers to monitor cumulative free-living activity in patients with COPD.

A team at MIT introduced a mobile telemetric continuous health-monitoring device that can continuously monitor variations in the patient's heart rate and blood oxygen saturation (Asada, Shaltis, Reisner, Rhee & Hutchinson, 2003). In addition, Waterhouse (2003) developed a wearable EEG device to monitor seizures in patients; the device consisted of several electrodes attached to the patient's head, with a recording unit placed on a belt or carried in a pouch.

Ward et al. (2006) used microphones and three-axis accelerometers mounted on the user's arms to identify activities in a wood workshop. Hanai, Nishimura and Kuroda (2009) used a 1D Haar-like algorithm filtering technique that is proposed as a feature extraction method for

a 3D accelerometer, to identify activities. In their work, they achieved a 93% accuracy in identifying some posture activities, namely walking, running, standing, ascending, and descending.

Gupta and Dallas (2014) presented an activity recognition system that uses a reading from a single waist-mounted triaxial accelerometer to classify gait events into six posture activities: run, jump, walk, sit, sit to stand, and stand to kneel to stand. The system used Relief-F and sequential forward floating search (SFFS) as the feature selector, and a classifier to identify the activity. The overall accuracy in identifying six posture activities achieved in this work was 98% when tested on the researchers' dataset. Brodie et al. (2015) also used a wearable device, a pendant, to study octogenarians ascending stairs. Their study achieved an accuracy rate for all the correctly identified events (stairs and not stairs) of 99.8%.

Gia et al. (2016) proposed an internet of things (IoT) fall detection system, which they claim is more energy efficient compared to similar solutions, being based on a customisable wireless sensor network that works for up to 35 hours without needing to be charged. In this work, the researchers combined wearable sensors with IoT by using a 3D accelerometer sensor, microcontroller, and wireless communication module. HMM was used to detect falls. The significant contribution of this paper was in creating a wireless sensor network that consumes less power.

Aziz, Robinovitch and Park (2016) used triaxial accelerometers and gyroscopes to differentiate between three states: walking, transferring, and sedentary (sitting). According to the authors, an accuracy of 100% was achieved when tested on their dataset, which was gathered from 10 young adult participants. Yogesh (2017) used the Shimmer2 wearable sensor data to identify activities, using three-dimensional tri-accelerometer data generated by Shimmer2. Different classifiers were used to identify the activities with 100% accuracy, according to the author, when using the random forest classifier.

Another approach by Huang et al. (2017) used triaxial accelerometers worn on both wrists and the waist to collect motion data that was used for activity recognition via machine learning techniques. The aim of this study was to show the impact of the location and the number of sensors used on the performance. The multi-sensor approached achieved 81% accuracy. The work also showed that performance depends on the location, as the dominant

location performed 7% better than the non-dominant location in identifying five posture activities.

Ejupi et al. (2017) used a wearable pendant device that contained a triaxial accelerometer and a barometric air pressure sensor to detect sit to stand movements using a wavelet-based algorithm. The authors tested the system on readings from 119 participants and achieved 93.1% sensitivity and a false positive rate of 2.9% for the sit to stand movement.

Perumal et al. (2017) suggested an IoT activity recognition system using wearable sensors, the Elgar framework and a Naïve Bayes classifier. The authors used Wi-Fi to connect the IoT nodes to the central server but only identified four simple activities: climbing, walking, sitting, and running. The achieved accuracy was 72%.

Mobark, Chuprat and Mantoro (2017) identified complex activities from accelerometer readings using hierarchal activity classification and labelling. The readings from mobile accelerometers were gathered into feature extractors, and the collected data was processed by extracting the mean and standard deviation features. Then, a KNN classifier was used to identify the activities. To identify an activity such as preparing breakfast, the person needed to perform some low-level activities (sitting and standing), then some everyday activities (lying down, getting up, and retrieving an object). From these readings, the classifier could identify the activity. The achieved accuracy for preparing activities combined was 90% when tested on the authors' dataset.

Based on the research reviewed in this section, the main advantages and disadvantages of wearable sensors can be summarised as follows.

The main advantages of wearable sensors are:

- They are not attached to an environment, which gives them the flexibility to be mobile and to be easily used in an indoor and outdoor environment.
- Different varieties of sensors can provide different readings, such as heart rate, blood pressure, accelerometers for activity recognition, and body temperature.

The main disadvantages of wearable sensors are:

- They identify only a limited number of activities, and to identify more activities, more sensors are needed.

- The location of the wearable sensors will affect the performance. This means the performance will vary from one person to another, as people do not share the same dominant parts (Huang, Yi, Peng, Lin & Huang, 2017).
- Such devices invade the detected person's space and privacy, as the sensors need to be in contact with the monitored person's body. In addition, they can sometimes cause an allergic reaction.
- Active sensors need to be charged, otherwise they will not be able to provide readings or will provide inaccurate information.
- There is a risk of losing the sensor.
- There is a risk that the person will forget to wear the sensor.

2.4.2 Environmental Sensors

Chen et al. (2015) proposed an activity recognition system for smart homes, using machine learning and streaming sensor data. They used two 2009 sensors layout from the CASAS dataset (Cook, Schmitter-Edgecombe, Crandall, Sanders & Thomas, 2009) to train an SVM classifier. The accuracy achieved in this study was modest, with an overall accuracy of 66% in identifying activities.

Mozer (1998) installed multiple sensors inside a house and developed a system called ACHE, which stands for Adaptive Control of Home Environment. ACHE has two objectives: the first is predicting the occupier requirements for temperature, ventilation, and lighting; the second is to save energy. Using neural networks, the system keeps updating itself when the user changes the temperature or the light. The goal of Mozer's (1998) work was to use sensor readings and neural networks to create a smart home that programs itself by observing the lifestyle and desires of the inhabitants, and learns to anticipate and accommodate their needs by controlling the heating, lighting, ventilation, and water heating. Mozer's (1998) system, however, is limited, and can only switch on/off a few items in the house, and the author provides no detail on how well the system performs.

Helal et al. (2003) define the concept of a smart home as "a home that can proactively change its environment to provide services that promote an independent lifestyle for elderly users". The researchers presented a concept for a smart home that contains different types of sensors installed in multiple locations in the house, such as doors, blinds, wardrobes, the

floor, and the bed. These sensors create a smart environment that is aware of all the objects that surround the person, and create mapping between the actual physical environment and the remote smart services (surveillance, action). This approach is considered one of the first steps towards realising the smart house concept. However, this approach is not easy to implement, as it requires modification to the house, and cannot be installed in all areas of the home.

A load cell that monitors the elderly while sleeping is presented by Adami, Hayes and Pavel (2003). The load cells are placed under the bed of the monitored person and provide readings to the system, which determines the sleeping characteristics and position in the bed of the monitored person. However, installing an active load cell under the bed is an expensive and inflexible approach. Also, only four positions in the bed were identified, with a 91% accuracy: lying on the back, on the left side, on the right side, and sitting.

Creating a smart home in a real-world scenario is hard to implement, as the house needs to be redesigned and modified to make it suitable to accommodate all the sensors, actuators, and controllers required. This can be extremely expensive, and will not be applicable in all homes (Cook & Das, 2007). Also, the maintenance is expensive as measurements are collected at many points in the house, which makes it harder to collect data from all locations and to power all the sensors; as such, a different solution is preferable.

Mehr, Polat and Cetin (2016) compared and tested three different artificial neural network algorithms to identify indoor activities: Quick Propagation (QP), Levenberg Marquardt (LM), and Batch Back Propagation (BBP). The three algorithms were tested on the Massachusetts Institute of Technology (MIT) smart home dataset (Tapia, Intille & Larson, 2004), and the LM algorithm achieved 92% accuracy in identifying activities. However, the researchers presented the results without any explanation of the reasons that led to the results.

Raeiszadeh and Tahayori (2018) proposed an activity mining and tracking method called the Uncertain Pattern-Discovery Method. Their study modelled the activities and behaviour of the monitored people to find interesting patterns and then used the frequent sequential pattern mining algorithm (FSPMA) and longest common subsequence (LCS) with a random forest classifier to identify the activity. The work was tested on the MIT dataset (Tapia, Intille

& Larson, 2004), and the average achieved accuracy in identifying 13 activities was 94%. This work was based on sensor readings and actuators, and the accuracy could be improved.

Ghosh et al. (2018) focused on identifying group activities from ultrasonic sensors. Using HMM and the ultrasonic sensors, they achieved an average accuracy of 89%. However, only three simple posture activities – walking, standing, and sitting – were recognised using their proposed approach.

Rimminen et al. (2010) presented a fall detection technique using near-field communication, pattern recognition, and combined pressure-sensitive floor sensors with a video feed from multiple cameras. They detected falls using near-field imaging (NFI), floor sensor, and pattern recognition, and estimated the pose using Bayesian filtering. The study achieved accuracy of 90%, sensitivity of 91%, and specificity of 91%.

Daher et al. (2016), presented a fall detection system using force sensors and tri-axis accelerometers concealed under intelligent tiles, combining the readings from the force sensors with the accelerometers with the aim of resolving the confusion between falling and lying down postures. The merged data was fed into a trained classifier that detected the posture. The system achieved an average sensitivity of 94% based on the provided training data. Both Rimminen et al.'s (2010) and Daher et al.'s (2016) systems are expensive, have problems with scalability, and require future-proofing as they require modification to the floor, cannot be fitted in all homes, and the accuracy could be improved.

Other researchers have found that using orientation sensors is not sufficient to provide all the required information to identify activities, and some of them are not accurate. Also, some sensors require calibration over time, which is not possible in all situations. Therefore, researchers have tried different approaches and used vision sensors.

Vision sensors provide more details about the environment compared to other sensors, such as colours, details about the location, and the shapes of objects in the area. Cameras are vision sensors that are used extensively in research. Different camera models can be used, such as the standard RGB cameras, and RGB-D depth cameras.

Researchers have achieved promising results from image sensors, and so have attempted to use various approaches and combine different sensors. The use of depth cameras is one of the most common approaches, as it adds an additional dimension (depth), which leads to more features that can be used. The invention of the Kinect sensor was a breakthrough in depth cameras. Kinect is one of the most popular depth cameras and has been extensively used for activity recognition as it provides a high level of accuracy with user-friendly software development kit (SDK).

Kurakin, Zhang and Liu (2012) used the Kinect to identify hand gestures. Their proposed system consists of five steps: segmentation to find the hand; tracking and filtering to find the hand regions; orientation normalisation to identify the hand directions; feature extraction; and classification. The average accuracy was around 80% in identifying 12 hand gestures when tested on the researchers' dataset. Liu et al. (2014) combined a depth camera with a wearable inertial sensor for motion tracking to be used in hand gesture recognition. The authors used nine generated signals from the Kinect camera (three signals), the triaxis accelerometer, and the triaxis gyroscope (six signals); these extra features, with the use of the multi-Hidden Markov Model, were able to achieve an overall accuracy of 91% when tested on the authors' dataset.

The introduction of the Kinect V2 with heart rate detection allowed researchers to use the camera for predicting heart attacks (Patel & Chauhan, 2014). A message can be sent, or a video conference can be started with a doctor or nurse, from the Kinect V2 based on the heart rate monitoring system. However, the patient must be facing the Kinect sensor within a fixed distance to enable the sensor to measure their heart rate.

Banerjee et al. (2014) proposed using a webcam combined with an IR sensor and Kinect sensor to identify activities. The approach was able to detect falls and fall-related activities based on fuzzy clustering. A Kinect sensor was used for activity segmentation and background subtraction, with Gaussian Mixture Model used for detection; the resulting silhouette was then supplied to the activity segmentation system. This study identified only a small number of activities: sitting, standing, and falling.

The problem with RGB-D cameras compared to RGB cameras is that they are not easy to install and maintain; for example, the Microsoft Kinect connects via USB and not via ethernet or WiFi. The covered area is also less than a regular camera. Researchers have stopped using this

type of camera for pose detection, as the standard camera is able to detect human poses (Wei, Ramakrishna, Kanade & Sheikh, 2016; Cao, Simon, Wei & Sheikh, 2016). In addition, the price of the standard camera is lower than the depth camera, and it does not require any specific software.

One of the most common uses of camera systems is face detection and recognition. Nefian and Hayes (1998) used a camera system with HMM for this purpose, with HMM used to construct the facial features and, based on the transition state probability, identify the face. However, their method works only with black and white images; furthermore, the eyes must be open and the person looking directly toward the camera. The method was tested on a small dataset of 48 images in total and achieved 90% accuracy.

Other researchers have used cameras to detect and track people, with the ability to detect and track more than one person being the focus of many research studies. Haritaoglu, Harwood and Davis (2000) created the W4 system, which works with black and white images and in an outdoor environment, using a single standard camera. First, the system scans the background and, based on foreground subtraction, identifies new objects in the images. It can then distinguish humans from other objects by using shapes and periodic motion cues. The W4 system can recognise action and differentiate between humans based on head detection and person segmentation, posture analysis, and body part detection. The tracking operation is based on silhouette and body part detection, and works by matching the template and predicting the motion, employing a tracking algorithm to “estimate the position of the torso of each human” (Haritaoglu et al., 2000). W4 was able to achieve good accuracy in tracking, detecting, and recognising some posture activities (standing 96%, bending 80%, lying 90%, and sitting 82%) based on information gathered from 170 silhouettes. It can also identify if the person is carrying an object in their hands. To detect the face, W4 combines two methods that shape cues and the vertical projection.

People tracking and localisation was also employed by Bahadori, Grisetti, Iocchi, Leone and Nardi (2005). The researchers used a stereo vision camera to track multiple people based on three data items from the stereo camera image intensity, disparity, and 3D world location to reacquire short-term human occlusion. The system was able to determine the person’s location with a less than 10% error rate, approximately 90% accuracy; however, the error rate increased with increased distance between the camera and the monitored person.

Fleck, Busch and Straßer (2006) used a smart cameras system to track multiple people using a particle filter and a colour histogram based on the particle filter. The main advantage of this technique is that all processing happens within the camera and hence, does not require a large bandwidth for data. For the same reason, it also provides high privacy; however, the system is more expensive compared to standard cameras and, as the tracking is based on colours, the performance is affected by the light conditions.

As researchers recognised that one camera could not cover all of the space in the monitored environment, they tried different approaches to cover a greater area; one solution is to use multiple cameras. Qian, Ma, Dai and Hu (2008) studied tracking multiple people with multiple cameras by using a particle filter and a colour histogram, using the colour histogram to detect humans. The primary concern with this approach is that, when the number of people increases, the number of particles will also increase, requiring greater computational resources. Also, accuracy depends on the colour of people's clothes.

In 2009, a study suggested building a system for supporting the elderly using a camera system that could measure their activities based on silhouettes via background subtraction and colour enhancement to extract the silhouette, with an SVM classifier (Harvey, Zhou, Keller, Rantz & He, 2009). As the system is based on using silhouettes, it has fewer privacy issues; however, it only works with one person and does not have tracking ability.

Zhou et al. (2011) created an automatic monitoring system for the elderly using standard cameras. Their system identifies activities based on silhouette extraction, infrared (IR) images, and supervised machine learning techniques. In their study, the authors used a live stream at night time, based on a supervised learning algorithm, and then improved the image with the median filter; they tracked the silhouette with a Kalman filter and SVM classifier. The study was able to focus on the identification of seven human postures: lying on the bed, standing, walking, sitting, falling, sitting on the bed, and falling from the bed. The advantage of using silhouettes is that it provides privacy for the monitored person. The main limitation of this study is that it used separated IR lights installed on a standing lamp, and only works with one person. Also, the accuracy of the overall system depends on the accuracy of the background subtraction and silhouette extraction. The achieved average accuracy was 93.25% when tested on the researchers' data.

Yu et al. (2014), installed a pan-tilt-zoom (PTZ) camera on a wheelchair and used this camera to identify humans based on facial characteristics and provide the details on an attached tablet with a touch screen. The tracking and detecting were based on face colour. The system can neglect non-facial information, using cascade Adaboost for facial feature matching, and HAAR-like face features for benchmarking. The study was a subproject of a smart wheelchair design project, and the goal was to integrate the face data processing with the mechanical control for the wheelchair.

Meinel et al. (2014) used a field-programmable gate array (FPGA) and omnidirectional smart camera system to track multiple persons. The authors presented this as an automated system for assisted living that worked in one location, and used foreground detection with blob extraction to track the monitored person. They used density-based spatial clustering of applications with noise (DBSCAN) to cluster the data, with DBSCAN and Kalman-GNN (global nearest neighbour) for tracking. The fisheye lens camera used in this system is more expensive than a standard camera and, although it covers 360 degrees, it needs to be in or close to a central location to cover all areas. Thus, the camera cannot be installed in all locations, such as corners. Also, the FPGA card increases the cost of system, and the foreground blob detection is affected by change in lighting, which adds extra noise to the detection.

In another study, Kolekar and Dash (2016) used a standard camera system with HMM and a classifier. The system merged optical flow and shape features, and HMM and SVM were then used to identify the activity. HMM achieved better accuracy results compared to SVM in identifying the activities when tested on the KTH (Schuldt, Laptev & Caputo, 2004), and Weizmann (Blank, Gorelick, Shechtman, Irani & Basri, 2005) datasets.

Fleck and Straßer (2008) used a smart camera monitoring system in a care home, to detect and identify humans based on colour and colour conversion from RGB to HSV. This study also used the colour system in camera handover. Based on the SVM algorithm, the system successfully predicted the activities, including fall detection. The main limitation in this study is that the system only detects one person, with low accuracy, and uses colour histogram, which means it is not efficient at night and when working with black and white videos.

To extract more useful features, researchers have tried to combine more than one sensor model to create a better system. One group of researchers focused on tracking and identifying the elderly in a care home setting using both audio and video features (Hauptmann, Gao, Yan,

Qi, Yang & Wactlar, 2004). The tracking was achieved by background subtraction and a mean-shift tracking algorithm. In addition to using colour to distinguish between people, assuming that they will not change their clothes, this technique is limited by its low accuracy when tested on the authors' work dataset, and has problems tracking people that partially occlude each other.

An activity monitoring and object recognition solution for a care home based on smart cameras and sensors was presented by Williams, Xie, Ou, Grupen, Hanson and Riseman (2006). The system is expensive, and is based on a fixed environment, however it consists of people tracking, fall detection, an alarm system, and an object finder.

Some studies have focused on converting the real-time video into a silhouette that can be read and understood by a caregiver to improve privacy, using sensors and cameras (Williams et al., 2006).

Below is a summary of the advantages and disadvantages of environmental sensors.

The main advantages of environmental sensors are:

- Non-image sensors, such as ultrasonic and gait recognition sensors, provide better privacy compared to image sensors.
- Image sensors are more flexible and expandable than non-image sensors.
- Images sensors provide more features compare to non-images sensors; therefore, they can be used to identify a wider range of activities.
- Different varieties of sensors provide different features, such as depth images, heat vision, and ultrasonic sensors.
- There is no risk of losing the sensors, as they are attached to the environment.
- As they are attached to the environment, they provide more freedom to the monitored person by not invading their personal space.
- There is no risk of the person forgetting to wear the sensor.

The main disadvantage of environmental sensors are:

- They provide generally low privacy.
- Non-image sensors provide limited readings compared to image sensors.
- Non-image sensors are less flexible than image sensors.

- They are attached to an environment and are thus immobile, which makes them less flexible.
- Some environmental sensors are expensive, such as heat vision cameras.
- They are active sensors that need to be attached to a power source.
- They require more power to work compared to wearable sensors.
- The location of the sensors will affect their performance, which means the performance will vary from one sensor to another, based on, for instance, the person's distance from the camera, and the viewing angle of the camera.

2.5 Data Processing

This section focuses on the technologies and algorithms that previous researchers have used in their work, such as HMM and NN, and the results achieved, based on the algorithms and the type of dataset used.

2.5.1 Hidden Markov Model

The Hidden Markov Model (HMM) is a prevalent method that has been extensively used in the last 30 years within different approaches, such as activity and speech recognition. It is one of the most important machine learning techniques.

Yang et al. (1997) proposed a straightforward framework to learn human actions using HMM representation, and applied it to two issues: human gesture recognition and skill learning. Their proposed approach focuses on classification and modelling of human activities, and is considered one of the early works on human action recognition using HMM.

Wang et al. (2007) introduced a features descriptor for the human silhouette. The low-level features are represented by radon transform and R transform, then a set of HMMs are trained to recognise the activity, as shown in Figure 2. 9. In their work, only four simple activities were identified when tested on their dataset: rush, carry, bend, and walk.

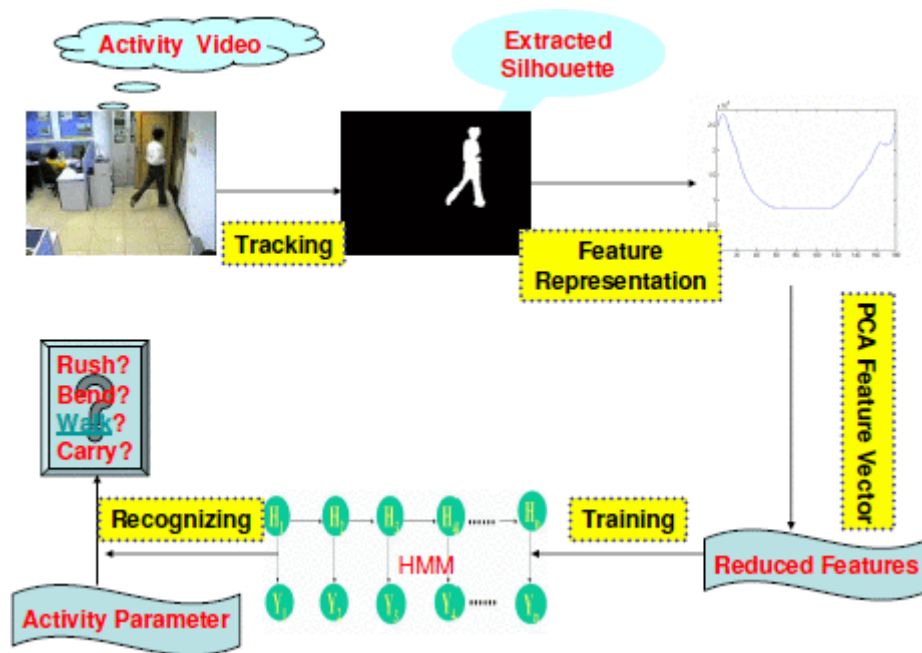


Figure 2. 9: Flowchart for the activity recognition system (Wang, Huang & Tan, 2007).

A cheap fall detection system using one Kinect sensor is presented by Dubois and Charpillat (2013). They used a running average to extract the background from the video images, and used the running average to learn the background by finding the average distance overtime for the depth map. They then used the data to train the HMM to identify eight different posture activities: squatting, lying on the couch, sitting, falling, lying down, walking, going up, and bending. The average sensitivity achieved was 86.25% when tested on their small dataset.

Wu et al. (2014) suggested processing Kinect sensor readings using SVM and HMM to identify activities in different indoor areas. Their proposed approach combines the SVM classified motion features, body structure features, and polar coordinate features with the HMM model to identify activity. The work was tested on a 3D daily activity database and achieved more than 95% accuracy for 12 different activities. There were some issues with the method; the first is that new errors were added to the method from the two steps of training for the SVM and the HMM. Each training step adds error, meaning it is not easy to find a system that achieves a 100% performance.

Another group of researchers used multiple HMMs for classification, by merging the data from three different HMMs and three different types of sensor to identify hand gestures (Liu, Chen, Jafari & Kehtarnavaz, 2014). Liu et al. (2014) used wearable sensors (an accelerometer and gyroscope) and a depth camera (Kinect) to recognise hand gestures. Each of these sensors

had its own HMM; the three HMMs were then merged in a pooling system, which merges the outcomes of the classifiers. Based on the pooling step, the system will identify the gesture, as shown in Figure 2. 10. Pooling the three HMMs together improved the accuracy by 7% over using a single HMM, and the authors achieved 91% accuracy in overall gesture recognition when tested on their dataset. There were two training stages in this work, each one of them adds some error to the model, and four training models for each hand gesture, which may introduce more error to the performance.

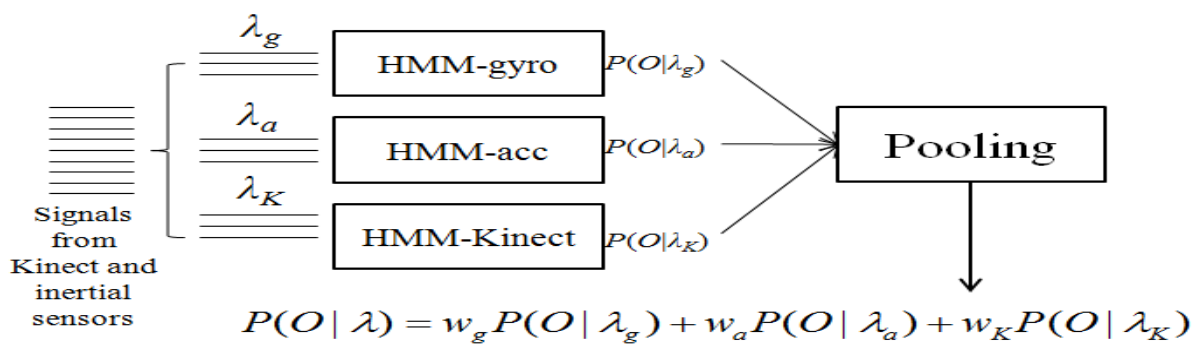


Figure 2. 10: The merging of three different HMM, $P(O|\lambda)$ is computed based on the three likelihood probabilities $P(O|\lambda_g)$, $P(O|\lambda_a)$ and $P(O|\lambda_K)$. (Liu et al., 2014).

Jalal et al. (2015) developed a system that creates a depth map for the human silhouette, which is used to separate the detected human from the environment. The detected silhouette features are fed to the k-means clustering algorithm; then, the data goes to an HMM, shown in Figure 2. 11. Jalal et al. (2015) achieved an 83.92% accuracy when the system was tested on the MSRAction3D dataset.

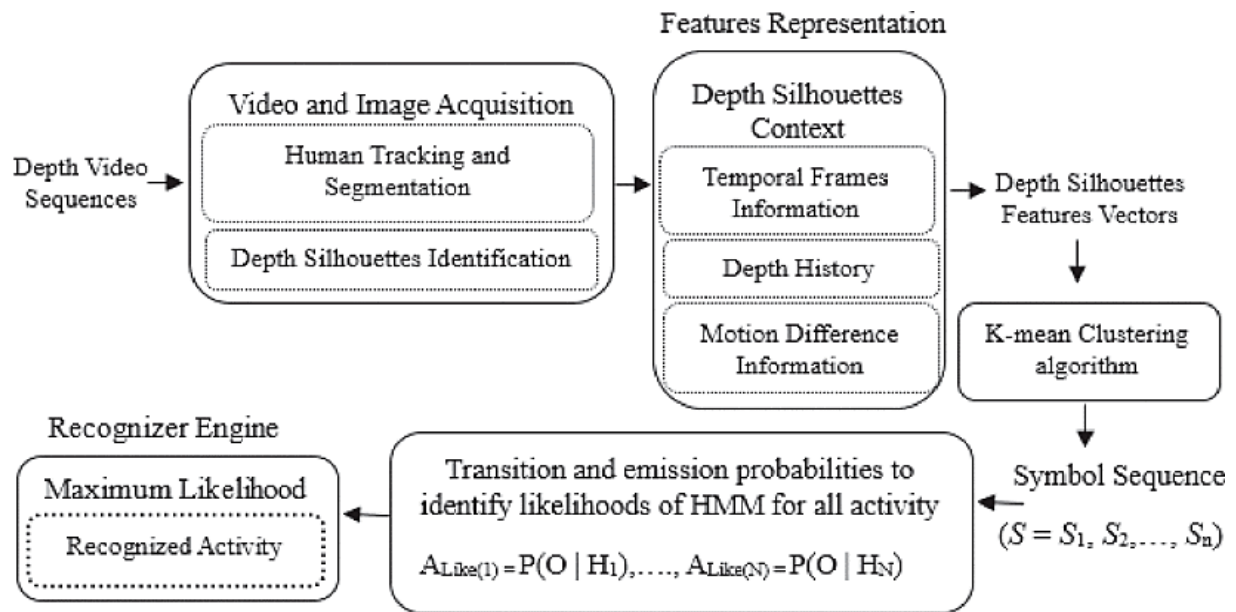


Figure 2. 11: Jalal et al. (2015) human activity recognition system (Jalal, Kamal & Kim, 2015).

A Kinect was used by Gaglio et al. (2015) to identify human activities using 3D posture data. In Gaglio et al.'s (2015) work, three different machine learning techniques were used to determine the postures and predict the activities: K-means clustering, SVM, and HMM. Their system consisted of three parts: the features extraction, then the posture analysis, and finally activity recognition, as shown in Figure 2.12.

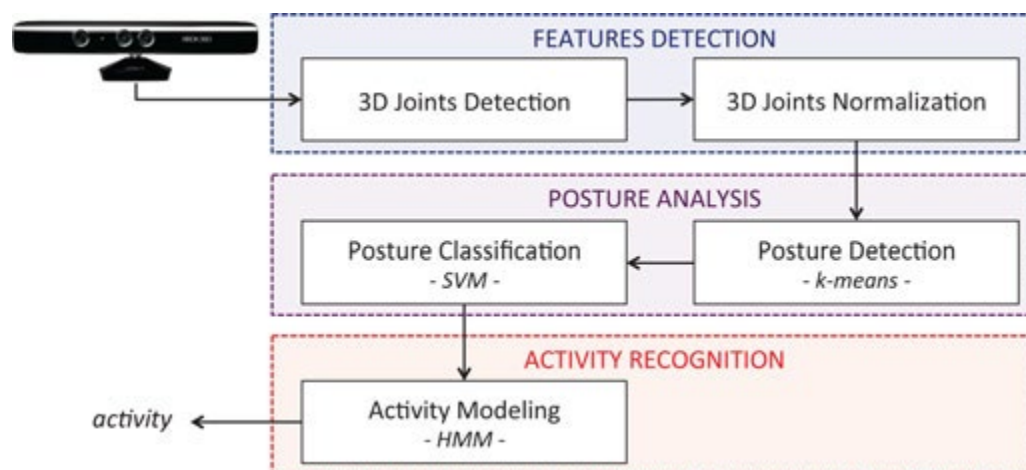


Figure 2. 12: Activity recognition from Kinect (Gaglio, Re & Morana, 2015).

Initially, the data from the Kinect is put through K-means clustering and then sent to the SVM classifier to identify the human posture. After that, the SVM output is sent to the HMM to identify the activity. Different datasets were used to test this system: the Kinect Activity Recognition Dataset (KARD), an activity data set created by the authors; and the Cornell

Activity Dataset CAD-60 (Sung, Ponce, Selman & Saxena, 2012). The achieved precision and recall were 77.3% and 76.7%, respectively; however, the system identifies posture activities only, and the three stages of learning may add error to the system's total performance.

Uddin et al. (2016) used body part histograms and HMM. The data from a depth camera was segmented using a random forest classifier; the histogram features for each body parts were then combined and applied to an HMM. The proposed system is shown in Figure 2. 13. The study focused on six simple posture activities and achieved a 94.5% recognition rate.

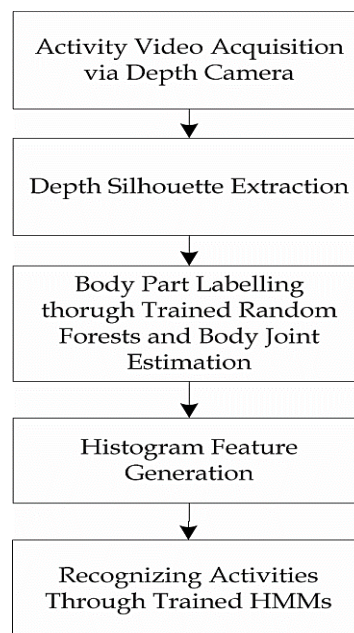


Figure 2. 13: Block diagram for the activity recognition system from a depth camera with HMM (Uddin et al., 2016).

Tsai, Ou, Sun and Wang (2017) presented 3D skeleton joint detection system using Kinect, which identified activities based on 3D pose data from the Kinect sensor using HMM. The trained HMM identified activities from the pose measurement, and the system achieved an accuracy of 95.64% in identifying 11 activities: sitting, standing, walking, drinking water, talking on the phone, reading, stretch, akimbo, and follow me. The main issue with determining actions from pose only is that more than one activity can share the same or a similar pose, which makes using pose as the main feature insufficient. Furthermore, not all the identified activities are ADL or IADL. In addition, the location of the Kinect was at a body height location, which makes it easy for people and other objects to block the view.

Another approach was suggested by Kijak et al. (2003), who used audio and video features to identify activities in a proposed tennis classification and segmentation method. The audio and video features were merged using a trained HMM, and the output was one of four options: first serve, rallies, replay, and break. By combining the audio and video data, the authors improved the accuracy by 7% and 10% compared to the use of the visual and audio features alone, and achieved a total accuracy of 78%. However, only a limited number of tennis activities were identified with a relatively low accuracy.

Niu and Abdel-Mottaleb (2005), presented an approach that used threshold and voting with HMM to segment and recognise activities. The segmentation was based on a sliding window with a low pass filter to level the voting curves, to achieve the segmentation and recognition results. Only a limited number of activities were identified in this work: walking, sitting down, standing up, and writing on a whiteboard. The system achieved 89% accuracy when tested on the authors' dataset.

Martins and Batista (2009) presented a two-stage classifying approach for facial recognition system. In their work, they used Active Appearance Models (AAM) and low dimensional manifolds with Laplacian Eigen-Maps (LE) to extract the face geometry. The features were then sent to the SVM classifier to identify expression changes, and an HMM classifier was used as a person-specific expression recognition identifier. The first stage SVM achieved 96.8%, and the average facial recognition when identifying seven facial expressions from images was 75.6%. The system involves two-stage training, and the performance needs improvement.

Cilla et al. (2009) used cameras, feature extractors, and HMM to identify activities, and Best First Search and Genetic Algorithms to select the features that improve accuracy. The selected features were sent to an HMM classifier. Only five posture activities were identified in this work: walking, bending, stopped, falling, lied down, and rising.

Uddin et al. (2010) employed a stereo camera and HMM to identify human activities, using the joint angles of the human body in 3D to determine the activity. The features from the joint angles were mapped into codewords to generate a sequence of Discrete-HMM. The system was tested on simple activities and achieved an average 92.5% accuracy. All of the

identified activities were posture activities: left-hand up-down, right-hand up-down, both hands up-down, boxing, left leg up-down, and right leg up-down.

Li and Fu (2012) presented an early human activity recognition system that modelled human activity as a time series. Treating human activities as a pattern of time series data, they proposed a new model, the Autoregressive Moving-Average Model (ARMA-HMM). HMM was used to predict the global pattern of the temporal observations of data, and ARMA was used to predict future values in a local range of time series. The authors also encoded the activity features as a subsequent observation of motion signals and called this the Histogram of Oriented Velocity (HOV). They tested the work on the UCF50 dataset (Reddy & Shah, 2013), and YouTube videos. The authors claimed that the proposed approach is superior to Discrete-HMM and Gaussian-HMM; however, they modelled only a single-person activity, and the method could not identify the activities of multiple people.

A study by Piquier et al. (2012) developed a recognition system based on wearable cameras and Hierarchical HMM (H-HMM). They proposed a two-level HMM for activity recognition, then set different levels of fusion to combine the audio and the video features. For each fusion, the researchers used leave-one-out cross-validation to evaluate the performance. The main concept of this work is the fusion of multiple features.

A dynamic SVM-HMM activity recognition system was proposed by Bansal et al. (2013), combining structural and temporal video sequence information. The work focused only on kitchen scenarios and identified activities based on features from context-based gesture recognition and image segmentation. An accuracy of 72% was achieved in identifying nine cooking activities from the Kitchen Scene Context-Based Gesture Recognition (KSCGR) dataset (Shimada, Kondo, Deguchi, Morin & Stern, 2013).

Another group of researchers focused on using feature extractors to identify features from videos; the researchers combined optical and shape-based features for human activity recognition and then grouped the features using the K-means clustering algorithm (Kolekar & Dash, 2016). Then, they modelled the clustered data into an HMM. Several different datasets were used in this work, including the Weizmann dataset (Blank et al., 2005), KTH Online databases (Laptev & Caputo, 2004), the Indian Institute of Technology dataset, and the Patna

dataset (Kolekar & Dash, 2016). The achieved average accuracy was 90%, according to the authors.

This section has demonstrated that different HMMs have been used by researchers and achieved high performance in many fields, such as voice and sound recognition, face recognition, and activity recognition. The performance of the HMM models differs from one dataset to another, and depends on the size and type of training data; consequently, the performance of HMM varies between 72% and 97%.

Although HMM can achieve an acceptable level of accuracy in identifying activities and action, there are some issues with the technique:

- The first issue is the performance, as the achieved performance can be improved, and is lower than the conditional random fields (CRF) and Deep neural network (DNN) (Krishnan & Cook, 2014).
- Another issue is expansion. As each activity is represented by an HMM, adding a new activity means creating a new HMM model for the new activity, which means the new model must be trained to identify the new activity.
- HMM mainly works with sequential data.
- Obtaining the correct amount of training data to create the emission and transition probabilities and avoid underfitting and overfitting is a concern.
- The HMM uses handcrafted features, and the quality of the features affects the model's performance.

2.5.2 Neural Networks

Neural networks are used in many applications, such as image classification, image recognition, and speech recognition. NNs are excellent feature extractors and selectors that perform well in learning high-quality features, i.e. the most useful features for the task (Wang, Zhang, Gao, Yue & Wang, 2017). More recently, researchers have begun to prefer deep neural networks (DNNs) over shallow neural networks, although some researchers (Ba & Caruana, 2014) have suggested that shallow NNs can perform as well as DNNs if they are trained in a way that mimics deep models. However, increasing the number of neurons is not appropriate in all cases.

The majority of the work conducted by other researchers and companies such as Alexnet (Krizhevsky, Sutskever & Hinton, 2012), GoogleNet (Szegedy, Liu, Jia, Sermanet, Reed, Anguelov & Rabinovich, 2015), VGG16 (Simonyan & Zisserman, 2014), and ResNet (He, Zhang, Ren & Sun, 2016), shows that the deeper the network, the better the performance, as illustrated in Figure 2. 14.

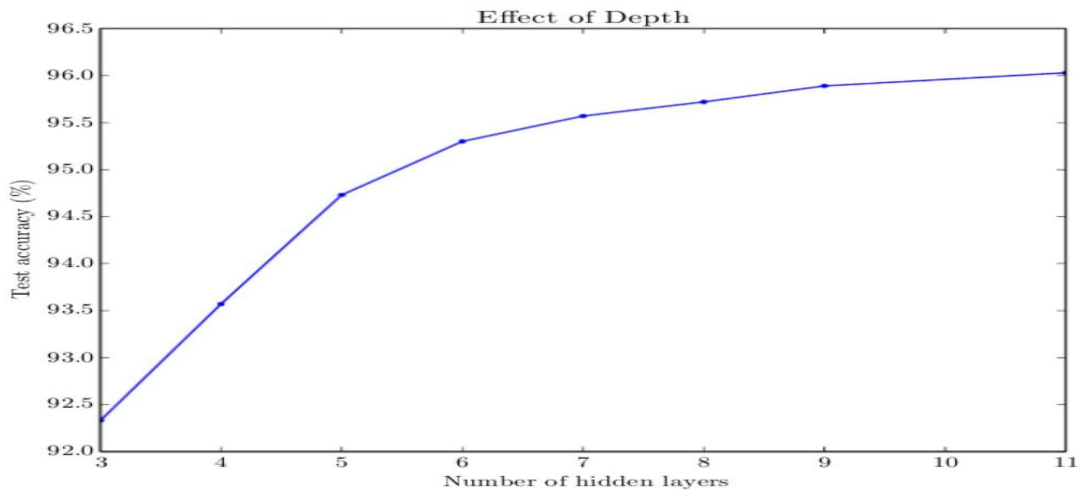


Figure 2. 14: The relationship between the depth of the network and the network accuracy (Goodfellow et al., 2016).

Cao and Nevatia (2016) made a comparison between two DNNs to identify activities in videos. They compare two streams of Convolutional Neural Networks and VGG with a classifier, where the two streams' eight-layer ConvNet gathered the features and identified the activity. The second proposed solution used a deeper NN VGG 16 that was trained on different data with an SVM classifier. The researchers found that the deeper NN with a classifier performed better than the CNN.

Researchers are working to achieve a fully automated recognition system that can learn to select features from the videos and images and identify activities. In addition, the advances in parallel processing, computing, and memory handling ease the use of DNN. Also, NNs show favourable results and are fast and efficient in almost all cases where they are used, provided they are configured correctly.

Convolutional Neural Networks (CNNs) are the most successful DNNs for image classification, detection, and segmentation (He et al., 2016). One of the oldest CNNs is AlexNet, a two-dimensional eight-layer CNN used to classify images (Krizhevsky, Sutskever & Hinton, 2012). The first five layers are convolutional, and the last three layers are fully connected layers.

Following the introduction of AlexNet, many attempts have been made to improve its classification/detection accuracy and efficiency. Researchers have worked on increasing the number of layers and making the deep network deeper, changing how the layers are connected, and changing the filter (kernel) size, such as the VGG network, which has 16 to 19 layers (Simonyan & Zisserman, 2014); GoogLeNet, which has 22 layers (Szegedy et al., 2015), and ResNet, with 152 layers (He et al., 2016).

Both NNs and DNNs have been used extensively in image recognition systems, and many researchers are now employing them in medical applications (Simonyan & Zisserman, 2014; Ravi et al., 2017) as they can achieve high performance at low cost with minimal human intervention. However, it is not recommended to use NN in all health sectors. As it requires a large amount of training data, each model is designed to perform a specific task, and can be misled to provide inaccurate results when noise is added to the image (Nguyen, Yosinski & Clune, 2015; Thys, Van Ranst & Goedemé, 2019).

Some researchers have adapted the popular NN configuration. For example, Wang et al. (2015) used two-stream deep CNN to identify activities in video files by adapting a similar architecture to GoogLeNet and VGGNet. The temporal feature that Wang et al. used in their work is to identify the activities is 10 frames. The model was tested on UCF101 and achieved recognition accuracy of 91.4%.

A segmentation approach using CNN is presented by Long et al. (2015). The researchers built a fully trained semantic segmentation using CNN, and transferred the learned representations from AlexNet (Krizhevsky, Sutskever & Hinton, 2012), VGG (Simonyan & Zisserman, 2014), and GoogLeNet (Szegedy et al., 2015) classification algorithms, and used these for image segmentation. Their approach achieved an accuracy of 65.4% on NYUDv2 (Silberman, Hoiem, Kohli & Fergus, 2012), 85.2% on SIFT Flow (Liu, Yuen & Torralba, 2011), and 62.7% on PASCAL VOC (Everingham, Van Gool, Williams, Winn & Zisserman, 2010).

A multi-stream image classification model using RNN was presented by Wu et al. (2015); their work integrated spatial, short-term motion, long-term temporal, and auditory clues in videos. The gathered features from the LSTMs were merged using adaptive fusion. This model was tested on UCF101 and Columbia Consumer Videos and achieved an accuracy of 92.2% and 84.9%, respectively.

Tran et al. (2015) trained a deep 3D ConvNet, which outperformed the 2D ConvNet. This technique used 3D convolution kernels and was able to learn temporal features in addition to spatial features. An action recognition accuracy of 90.4% was achieved when tested on UCF101 dataset.

Ma, Fan and Kitani (2016) used two DNNs and fused them in the last feedforward layer to identify activities, based on identifying appearance information and motion information. Their model was tested on three public datasets: GTEA, with an accuracy of 76.15%; GTEA Gaze (Gaze), with an accuracy of 55.55%; and GTEA Gaze+ (Gaze+), with an accuracy of 74.34%.

A group activity recognition system using RNN was presented by Ibrahim et al. (2016). A two-stage Long Short-Term Memory (LSTM) model was built to identify activities: the first stage identified the person's activity; the second stage identified the group's activity. The researchers tested the system on two datasets: the Collective Activity Dataset (Choi, Shahid & Savarese, 2011) and a new volleyball dataset. In this work, the two-stage hierarchical model achieved an accuracy of up to 81.5% on the Collective Activity Dataset, and 51.1% accuracy on the volleyball dataset.

Wang et al. (2017) used sensors to identify activities. The authors developed a wireless localisation and activity recognition system using a combination of unsupervised and supervised machine learning techniques. In order to learn the discriminative features automatically, a three-layer NN was used to get the features from the wireless sensors. The achieved accuracy was 85% and 88% for activity recognition and gesture recognition, respectively. However, this system could only identify eight posture activities.

Lee, Yoon and Cho (2017) proposed one-dimensional CNN for human activity recognition based on accelerometer data. The accelerometer data was transformed into vectors and then fed as inputs to the 1D CNN. The output was one of the three possible activities: walking, running, and staying still. This technique achieved an accuracy of 92.71%, but only identified three posture activities.

Ke et al. (2017) studied action recognition with a depth camera. The researchers identified human activities from skeleton image sequences using CNN. The first step was extracting the spatial and temporal features using CNN by transferring three sequences of the skeleton clip

to a new representation. The next step was to use a Multi-Task Learning Network (MTLN) (Caruana, 1997); the proposed MTLN consisted of a fully connected layer, *ReLU*, a second fully connected layer and a Softmax layer. The researchers used VGG19 as the feature extractor. After learning the features, the MTLN was used to join the feature vectors from CNN. Three datasets were used to evaluate the method: the NTU RGB+D dataset, the SBU Kinect interaction dataset, and the CMU dataset. The achieved accuracies were 84.83%, 93.57%, and 93.22%, respectively. The two training steps, CNN, and then MTLN, affected the results in this work.

An action recognition system using spatial and temporal features was presented by Feichtenhofer, Pinz and Wildes (2017). The two-stream CNN was trained to identify the action. The system was tested on UCF101 and HMDB51 datasets, and achieved accuracies of 94.9% and 72.2%, respectively.

A two-level hierarchy NN to recognise group activity was proposed by Shu et al. (2017). A modified version of LSTM, called the Confidence-Energy Recurrent Network (CERN), was used to determine group activities, human interactions, and individual actions. Two versions of CERN were presented: CERN-1 and CERN-2. CERN-1 used two LSTMs at the bottom level, and computed the individual action classes; while CERN-2 had an extra LSTM to compute group activity. Two datasets were used to test this work; for the Collective Activity dataset, the achieved accuracies were 85.5% for CERN-1, and 88.3% for CERN-2; for the Volleyball dataset, the achieved accuracies were 82.3% for CERN-1, and 83.6% for CERN-2. CERN-2 outperformed CERN-1, however, it increased the complexity of the system.

Huang et al. (2017) presented a skeleton action recognition based on the Lie group, a ubiquitous concept in mathematics: a smooth manifold, a specific kind of geometric object (Knapp, 2013; Savage, 2015). In their work, the model learned the Lie group features for 3D action recognition using DNN, by converting the Lie group features into DNN-friendly features using rotation mapping layers. The work was tested on three different datasets: the G3D-Gaming dataset, the HDM05 dataset, and the NTU RGB+D dataset. The achieved accuracies were 89.10%, 75.78%, and 66.95%, respectively.

Fei-Fei et al. (2006) used the idea of learning from others' work to develop a Bayesian learning framework for object categorisation. They presented a method that learns information about

a category from a single or a few images, and obtains some of this information from previously learned categories. The researchers named their method One-Shot Learning.

A hierarchical rank pooling with DNN for activity recognition was presented by Fernando, Anderson, Hutter and Gould (2016). The researchers designed an algorithm to encode the video sequence at more than one level, after splitting the clip into multiple overlapped parts. The first step was encoding the lowest level with rank pooling. Rank pooling creates a representation of the temporal features by learning how to rank the frame-level features of a video in chronological order. This will encode some of the temporal features. The second step was applying rank pooling again to the first layer to gather the higher-level features, as presented in Figure 2. 15. The researchers used Hollywood2, HMDB51, and UCF101 datasets, and the achieved accuracies were 76.7%, 66.9%, and 91.4%, respectively.

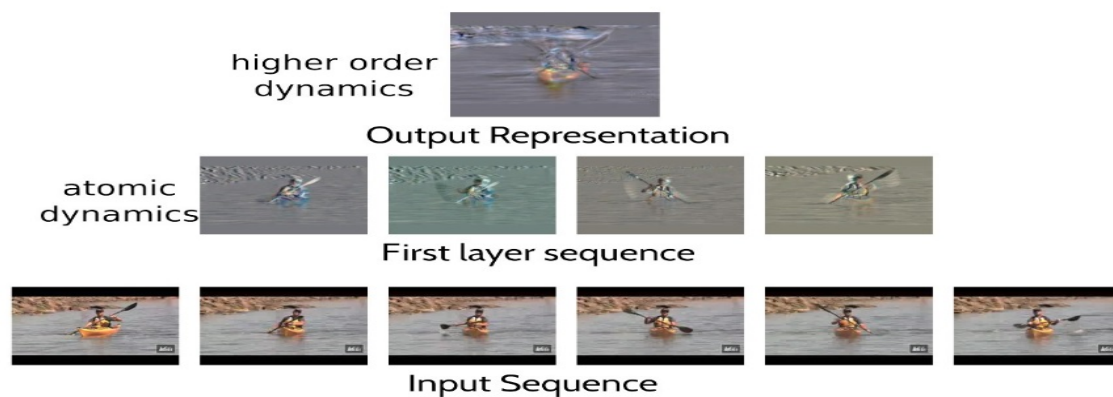


Figure 2. 15: The rank pooling hierarchical activity recognition system (Fernando et al., 2016).

The concept of using dynamic images for activity recognition was explored by Bilen et al. (2017). Temporal features were encoded using rank pooling, then CNN. This method first combined the temporal features with the spatial features, then it gets the features using CNN. The researchers used UCF101 and HMDB51 datasets, and the achieved accuracies were 89.1% and 62.6%, respectively.

There are several different models of ANN, such as Feedforward, CNN, and RNN. ANN has been used in multiple studies and has achieved accuracies ranging from 60% to 95% depending on the model and the dataset, as presented earlier. For example, when the UCF101 dataset is used, the performance using DNN varies between 89% and 94.9%, and for the HMDB51, performance varies between 62.6% and 72.02%.

Researchers have shown that, provided it is given a suitable amount of training data, the deeper the NN, the better the results, as researchers have found that the amount of training data determines the performance of the model. In general, NN requires more datasets for training compared to non-neural networks.

Artificial Neural Network technologies and algorithms have been shown to achieve higher performance compared to other machine learning techniques, as presented in this chapter. However, there are some issues with this method, which can be summarised as follows:

- ANN needs a large amount of training data to create a better model.
- It requires a high-end system to run the training step.
- Finding the optimum parameters for the NN model could be challenging.
- It is not recommended to use ANN in all situations, as it can be misled by noises and images, as proven by Thys, Van Ranst and Goedemé (2019).
- It is difficult to determine how the ANN achieves the results, due to the number of neurons, the way they connect, and their weights and biases.

2.5.3 Hybrid Approaches

Many researchers have used machine learning techniques to identify or classify objects. Selecting the best method is challenging, as it depends on the type of objects, number of samples, type of training data, amount of training data, and the required task. Wang, Takaki and Yamagishi (2016) compared three different machine learning techniques – HMM, DNN, and RNN – for their ability to discard noise in a text-to-speech conversion. They found that the system performance tended to increase with the increase in the amount of training data, primarily for the HMM and RNN. For DNN, deeper networks performed better.

Researchers have also found that combining different machine learning techniques, such as HMM and SVM, HMM and NN, and NN and SVM, improves performance compared to a single model of machine learning. For example, Rynkiewicz (1999) presented a hybrid HMM-MLP model for time series prediction, and achieved an improvement in segmenting the laser time series with the proposed hybrid design. Hybrid HMM-ANN models are widely used in text recognition (Bengio, LeCun, Nohl & Burges, 1995), biological applications (Baldi & Chauvin, 1996), speech recognition (Touretzky, Mozer & Hasselmo, 1996), digit recognition (Cosi,

2000), acoustic recognition (Stadermann & Rigoll, 2004), and activity recognition (Ordóñez, de Toledo & Sanchis, 2013). There are many advantages of using hybrid models over single models (Ordóñez, de Toledo & Sanchis, 2013), such as improving the performance and, in some models, requiring less training data. This helps to overcome the underfitting problem.

Riis and Krogh (1997) presented a hybrid framework for HMM and NN, called the Hidden Neural Network (HNN). The HNN improved accuracy by 8% compared to the HMM on the TIMIT acoustic-phonetic dataset (Garofolo, 1993), achieving an overall accuracy of 84%. However, their system suffers from complexity.

Dahl et al. (2012) proposed a hybrid DNN-HMM to perform large-vocabulary speech recognition (LVSR). They showed that DNN-HMM outperformed the normal GMM-HMM in speech recognition, and also required less training data. The concern, however, is the complexity of the system. Therefore, some researchers suggest using Feedforward NN and CNN with a larger amount of training data instead (LeCun, Bengio & Hinton, 2015). In addition to the complexity, Dahl et al.'s (2012) system used two machine learning algorithms (DNN and HMM), which creates the opportunity for adding more error to the system.

Li et al. (2013) investigated a different model of NN. Their approach was tested on two speech datasets, the eNTERFACE'05 database (Martin, Kotsia, Macq & Pitas, 2006) and the Berlin database (Burkhardt, Paeschke, Rolfes, Sendlmeier & Weiss, 2005). In their study, DNN-HMM with Restricted Boltzmann Machine (RBM) and DNN-HMM with discriminative pre-training achieved better results than HMM and GMM-HNN; the DNN-HMM with discriminative pre-training performed best. However, using the DNN to predict the states for HMM is complex, and the performance depends on the used dataset. When the method was tested by Han et al. (2014) on a different dataset, the DNN-HMM achieved the same results as the conventional HMM.

Ordóñez et al. (2013) presented a comparison between two hybrid approaches to recognising activities of daily living via wearable devices. In their work, they combined HMM with ANN, and HMM with SVM, to identify activities. They showed that the hybrid model achieved better performance compared to the single, non-hybrid model. In addition, HMM-SVM achieved better performance than HMM-MLP when tested on the Kasteren (Van Kasteren, Noulas, Englebienne & Kröse, 2008) and Ordóñez datasets. The Multilayer Perceptron (MLP) and SVM

were used to predict the posterior probability for the HMM. Seven daily activities were identified: leaving the house, using the toilet, taking a shower, sleeping, eating breakfast, having dinner, and drinking. They used three sensor models that provided readings about the status of doors (open/closed), IR sensors to detect motion, and float sensors to identify the toilet being flushed. The main limitation of this research was that the type of sensors used provide only specific information and cannot be used for another purpose.

Zhang, Wu and Luo (2015) proposed an HMM-DNN model to recognise activities from an accelerometer on a smartphone. They compared their model with HMM-GMM and HMM-RF (random forest), and used the DNN to model the emission probability, which was used to estimate the posterior probability. The HMM-DNN model outperformed HMM-GMM and HMM-RF models and achieved an accuracy of 93.5%. However, the work was tested on the researchers' own data that has not been published.

Other researchers have compared GMM, GMM-HMM, and DNN-HMM (Schröder, Anemüller & Goetze, 2016), and found that GMM-HMM performs better than DNN-HMM when tested on detection and classification of acoustic scenes and events (Schröder, Anemüller & Goetze, 2016). The researchers argued that the amount of data affected the results, being too low to properly train DNNs. In addition, the type of data and the designed model were essential parameters that affected the performance.

Each technology has its advantages and disadvantages. At present, there are no available systems that can identify ADL activities. Therefore, combining different models of machine learning techniques can help to improve the performance in specific aspects. Moreover, as each machine learning model targets a particular task, combining them might help to create a system that can achieve high accuracy in identifying ADL. However, combining these will increase the complexity of the system, creating more opportunity for error.

2.5.4 Pose Estimators

Kinects have been extensively used to identify human poses; this pose data can be used in different applications. For instance, Ramadijanti et al. (2016) used Kinect to create an app that can identify dance movements. Shen et al. (2014) used pose detection for pose correction and tagging. In their work, they use Kinect to detect the human skeleton, then normalised the Kinect values and used a regression function to predict the offset to perform pose correction. Finally, they used random forest classifier to predict the correct pose.

Duckworth et al. (2017) carried out an activity analysis using a system placed on a mobile robot, based on the idea of learning human activities from observation in an unsupervised way using Convolution Pose Machine (CPM), qualitative features, and K-means clustering. The researchers initially used the OpenNI tracker to detect the human pose, then replaced this with the CPM, as the CPM performs better than the OpenNI tracker. Kinect skeleton readings have also been used to identify activities by collecting readings for human body joints with deep LSTM (Zhu, Lan, Xing, Zeng, Li, Shen & Xie, 2016). Cippitelli et al. (2016) converted skeleton joint readings into feature vectors and then used multi-class SVM to identify the activities.

Takač, Català, Martín, Van Der Aa, Chen and Rauterberg (2013) used position and orientation tracking to locate humans and estimate their orientation. A smartphone was used as a wearable device to measure orientation using three-axial accelerometers, gyroscopes, and magnetometers, Kinect was used to provide the position information, and a classifier was used to classify the position and orientation. The system tested on their dataset that has 12 subjects. The issue with this study was that all the 12 participants were healthy, and the system intended to identify the position and orientation of patients with Parkinson's disease that suffer from the freezing of gait.

Other researchers have also used Kinect skeleton readings for human body tracking and pose estimation (Stommel, Beetz & Xu, 2015). The Kinect detects and tracks the skeleton of the monitored person, then converts the Kinect skeleton detection values into feature vectors and from those finds the pose. The estimation performance is affected by the person distance from the depth camera.

Pictorial structure has been used in many studies to construct human body parts (Felzenszwalb & Huttenlocher, 2005). The pictorial structure is an object description algorithm that represents shapes and objects. Initially, the algorithm estimates the overall appearance of the object; from that, it finds the local features that describe the appearance of the object. The final step is the spatial links between the local features (local appearance). In their work, Felzenszwalb and Huttenlocher (2005) used background subtraction that can be easily affected by light, and tested their work on black and white images. Vajda and Zoltán (2011) also presented a people detection and pose estimation system using pictorial structure. The pictorial structure was used as the appearance model, and was combined with

a body part detector based on a shape context descriptor and AdaBoost. The researchers tested the work on their own dataset, and claimed that it could improve detection accuracy and speed; however, their system cannot work in real-time.

Fei (2012) suggested combining the pictorial structure with image features to identify human poses in videos with different backgrounds. Pictorial structure with maximum likelihood was used to learn the model parameters and identify body parts using the canny detector as an image feature, and the colour histogram as a region feature. The achieved accuracies were 90% on Buffy (Ferrari, Marin-Jimenez & Zisserman, 2009), and 70% on PASCAL (Everingham, Eslami, Van Gool, Williams, Winn & Zisserman, 2015) datasets. However, Fei's (2012) method works only with images, can only identify the upper body of the detected person, and the performance needs some improvement.

Yang and Ramanan (2013) presented an articulated human detection method using pictorial structure and orient gradient descriptors. This work is considered one of the most successful attempts to represent body parts based on spatial features (Newell, Yang & Deng, 2016; Wei et al., 2016; Cao et al., 2017). A visualisation of this work is shown in Figure 2. 16, which presents pose detection based on 14 points. The work was tested on the Parse dataset and achieved an overall accuracy of 79%. On Buffy dataset, it achieved an accuracy of 88.5%. The most important limitation of the pictorial structure method is the difficulty differentiating between sides (cannot differ between left and right-hand sides).

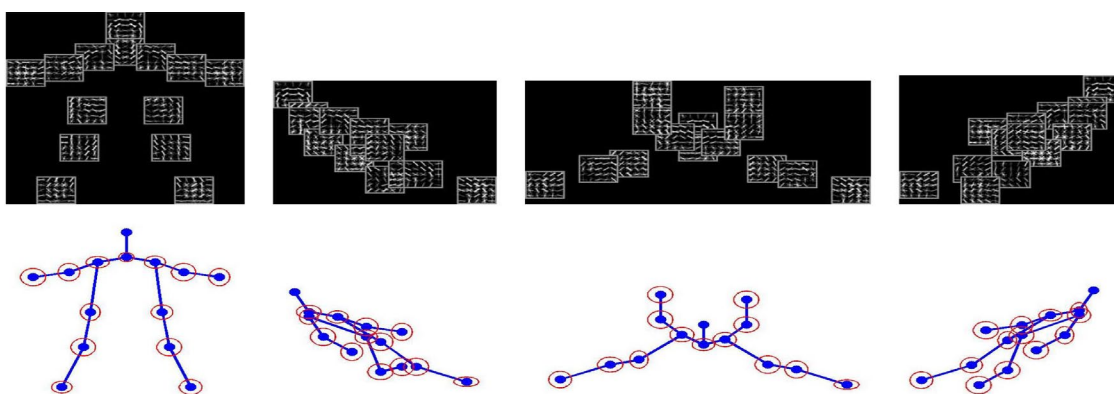


Figure 2. 16: The articulated human detection and human pose estimation (Yang & Ramanan, 2013).

An articulated pose estimation approach called the Pose Machine was presented by Ramakrishna et al. (2014). The Pose Machine is an algorithm that predicts visible body parts and uses scene parsing to gather the hierarchical estimation for the human body. The features

are gathered using Histogram of Gradients (HOG) features, lab colour features, and gradient magnitude; the features are then fed into a classifier. The achieved accuracy on the LEEDS Sports Pose dataset was 72% (Johnson & Everingham, 2011). The method consists of multi-pose machines, i.e. multi-stage training, and uses fixed handcrafted image features; it works only with images, and the accuracy needs improvement.

Youness and Abdelhak (2016) used Kinect with a machine learning algorithm to identify human poses. The Kinect reading provided the skeleton data, which gave readings for 15 joints points; the distance between the points was then calculated. After that, a classifier was used to identify the pose. Different classifiers were used, including Support Vector Machines (SVMs), Artificial Neural Networks (ANNs), the K-nearest neighbour, and a Bayesian classifier. According to the authors, SVM performed best. This method depends mainly on the difference between the positions of the arms.

Wei et al. (2016) presented the Convolution Pose Machine (CPM), a fast and accurate human pose estimator that uses ConvNet and belief maps to estimate human poses. The authors designed a CNN that worked on the belief maps: “The CPMs consists of a sequence of convolutional networks that repeatedly produce 2D belief maps” (Wei et al., 2016). The belief maps provided spatial information for the human body parts, and the ConvNet learned the links between the human body parts. The CPM framework is shown in Figure 2. 17.

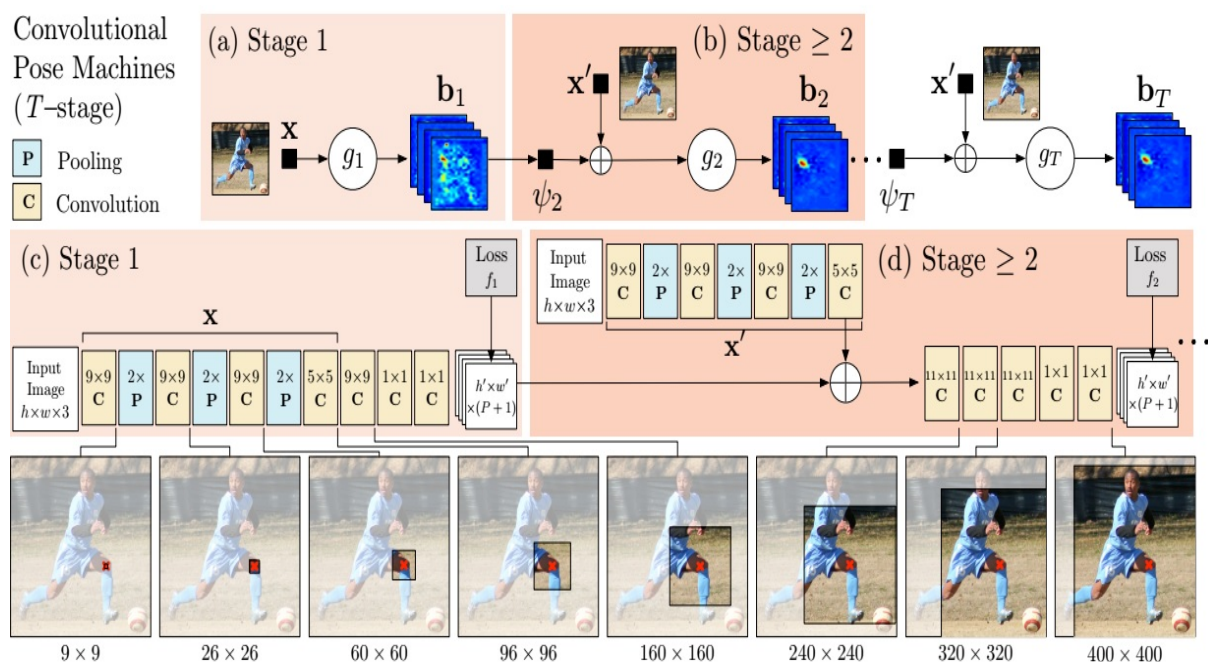


Figure 2. 17: The complete framework for the CPM (Wei et al., 2016).

The CPM is a trained class predictor that predicts the human body parts, and the convolution is trained with CNN; to produce the belief map, each stage of the pose machine is trained repeatedly. The model was tested on three datasets, the MPII (Andriluka, Pishchulin, Gehler & Schiele, 2014), the Leeds Sports Pose (LSP), and the FLIC (Sapp & Taskar, 2013). It achieved an accuracy of 90.5% when tested on Leeds Sports Dataset. The CPM improved the accuracy of the previous work by Ramakrishna et al. (2014), but had problems in detecting the body parts of occluded people.

Cao et al. (2016) presented real-time multi-person two-dimensional pose estimation using Part Affinity Fields (PAF). This is considered one of the fastest and most accurate human pose estimators using video cameras, and works in real-time. The researchers used two branches of multi-stages DNN; the first branch predicted the confidence maps, and the other branch predicted the affinity fields. The PAF uses a 2D vector field to encode the location and orientation of body parts over the image domain. It uses new feature representation, which saves the location and orientation information for a human body part. The work was tested on the MPII Multi-Person Dataset and COCO 2016 key points challenge, and the algorithm achieved higher performance than all other available techniques.

Park, Ji and Chun (2018) tried to improve the pose detection of the CPM for occluded body parts. Instead of using standard images as an input, they used RGB-D images and used the CPM to create the belief map for the human body parts. As they used a depth camera, the distance from the camera will affect the system's performance.

Go and Aoki (2016) tried to identify human poses from a top-view camera by using a flexible bounding box. They used a CNN called Bounding Box Curriculum Learning (BCL) and Recurrent Pose Estimation (RPE). The work was tested on the authors' own, small dataset; the performance in detecting poses from top-view images requires improvement.

Guo, He and Guan (2017) presented a pose estimation model using an RGB-D camera with CNN and LSTM. The CNN was used to get the information from the images; PoseNet was used as a baseline pose estimator; and the LSTM was used for temporal information. The researchers slightly modified GoogLeNet by discarding the pooling and used it as the CNN. The authors claimed that combining the CNN with the LSTM improved the accuracy of pose estimation when tested on the ICL-NUIM RGB-D dataset (Handa, Whelan, McDonald &

Davison, 2014). The model involved two training steps, one for the CNN and one for the LSTM, and works with depth cameras.

Liu and Ferrari (2017) presented an active learning algorithm to identify human poses, using the CPM with a small amount of labelled data and an active learner that labelled the important unlabelled samples to improve the model. The selected data was hand labelled. After that, the learner updated the CPM with the labelled data. This model achieved accuracies of 86.7% and 88.9% on the MPII human pose and the extended Leeds Sports Pose datasets, respectively. The most significant issue with this approach is the inability to handle complex scenarios or situations when noise affects the active learner; it is also sensitive to bias in the sample training data.

Bulat and Tzimiropoulos (2016) presented a CNN cascaded architecture model to identify human poses using a heatmap. Dual ConvNets were designed to learn the links between body parts and the spatial context, and then to find human poses even in cases of severely occluded parts. The proposed cascade ConvNet consists of two deep subnetworks; the first detects the individual body parts, for which the authors used a modified version of VGG-16, as they replaced the fully connected layers of VGG-16 with convolutional layers; the second is a regression model to link the parts to create the body heatmap. The researchers used ResNet as a base network and the model was tested on MPII and LSP datasets, achieving impressive results. However, optimisation is more difficult in Bulat et al.'s (2016) model as it contains a higher number of parameters. It is also designed to identify the poses of a single person only.

Multiple techniques have been used to identify poses using depth cameras and normal cameras. Algorithms, such as the CPM, help to improve the performance of pose detection using normal cameras. Using the pose features alone to identify activities has not achieved the best results in activity recognition and requires further improvement.

2.6 System Evaluation

This section presents the datasets and the metrics that previous researchers have used. Various different datasets have been used by researchers; some focused on using image datasets that can be used for object recognition and segmentation, such as the COCO dataset (Lin, Maire, Belongie, Hays, Perona, Ramanan & Zitnick, 2014), while others used video datasets for activity recognition, such as the KTH dataset (Schuldt, Laptev & Caputo, 2004).

To evaluate their models, researchers used different metrics, such as accuracy. This section highlights the most important datasets and metrics relevant to the work undertaken in this study.

2.6.1 Image Datasets

Researchers have introduced different types of image datasets that can be used for object or pose identification. Buffy pose classes (Ferrari, Marin-Jimenez & Zisserman, 2009) is considered one of the first pose image datasets, which that took data from the popular television series ‘Buffy the Vampire Slayer’ and is intended to test pose retrieval systems. The Leeds Sports Pose (LSP) dataset (Johnson & Everingham, 2011) is an image dataset that contains athlete poses across 10,000 images gathered from Flickr searches, and includes poses of athletes engaged in different sports. The dataset labels include the details for 14 human body parts (right ankle, right knee, right hip, left hip, left knee, left ankle, right wrist, right elbow, right shoulder, left shoulder, left elbow, left wrist, neck, and top of head). A sample of the dataset is shown in Figure 2. 18.

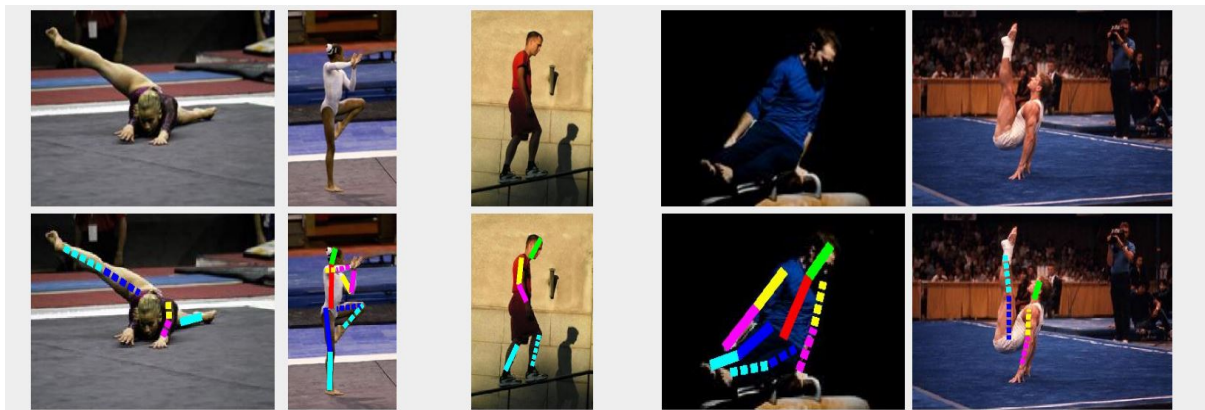


Figure 2. 18: Example from the LSP dataset.

The Human Pose Evaluator (HPE) is another image dataset (Jammalamadaka, Zisserman, Eichner, Ferrari & Jawahar, 2012). The images in this dataset were randomly taken from Hollywood movies and the ‘Buffy the Vampire Slayer’ television series. In this database, human actors with more than three body parts visible are annotated based on six upper body parts, in a stickman shape. These six parts are the head, torso, upper and lower arms, as shown in Figure 2. 19.



Figure 2. 19: Sample from the HPE dataset with upper body annotation.

The Frames Labeled In Cinema (FLIC) dataset is also an image dataset taken from famous Hollywood films, and provides information about the human pose. The dataset is annotated and labelled according to ten upper body joints (Sapp & Taskar, 2013). A sample of the dataset is shown in Figure 2. 20.



Figure 2. 20: Sample from the FLIC dataset.

ImageNet (Deng, Dong, Socher, Li, Li & Fei-Fei, 2009) is one of the largest image datasets, containing almost 15 million images, and has been used by many researchers to train their models. It is organised according to the WordNet hierarchy, and an average of 1,000 images are used to clarify synset (synonym set).

COCO is another large dataset that can be used for object detection, segmentation, and captioning (Lin et al., 2014). The COCO dataset contains 330,000 images, 1.5 million object instances, and 80 object categories with five captions for each image.

The PASCAL Visual Object Classes (VOC) dataset is labelled for classification, detection, and person layout (Everingham et al., 2015). This dataset contains classes for people, animals, vehicles, and indoor objects, and is widely used in image detection, segmentation, action

classification, and person layout challenges. The 2012 challenge for PASCAL VOC contains 20 classes, 11,530 images containing 27,450 ROI annotated objects, and 6,929 segmentations.

2.6.2 Video Datasets

This section discusses the available activity video datasets, which are categorised based on the type of cameras that were used to record the activities, standard camera (RGB) or depth camera (RGB-D).

2.6.2.1 Standard Camera Video Datasets

The KTH dataset (Schuldt, Laptev & Caputo, 2004) is one of the most popular datasets and has been extensively used in activity recognition. The KTH dataset offers six different actions: walking, jogging, running, boxing, hand waving, and hand clapping. In this dataset, 25 people in four different backgrounds perform the actions. The clips were recorded using the AVI file format, with a 160x120 pixel resolution. All the recordings are in black and white. A sample of the KTH human action dataset is shown in Figure 2. 21. The KTH dataset focuses on activities that require posture movement within a time frame, i.e. posture activities, which can be performed in any location and any direction.



Figure 2. 21: Sample from the KTH dataset.

The Weizmann dataset (Blank et al., 2005) is an action dataset similar to the KTH dataset. It contains 90 video clips that are 50 frames long with a resolution of 180x144 pixels. Ten actions were performed by nine subjects in front of a single camera: run, walk, skip, jumping-jack,

jump-forward-on-two-legs, jump-in-place-on-two-legs, gallop sideways, wave-two-hands, wave-one-hand, bend. A sample of the dataset is shown in Figure 2. 22. This dataset also focused on posture activities, and the actions were recorded in an outdoor location.

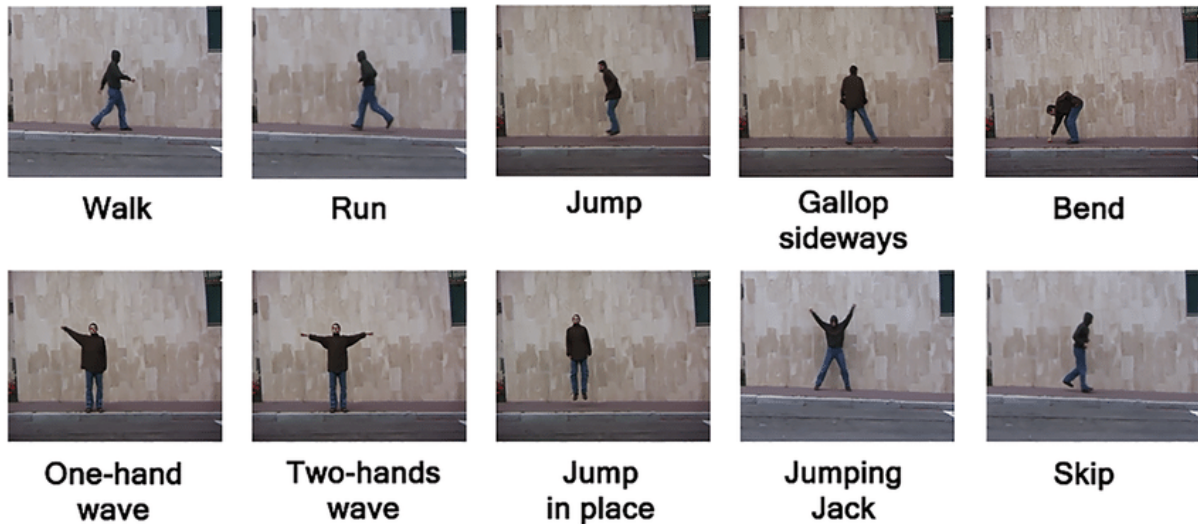


Figure 2. 22: Sample of the Weizmann dataset.

The TUM Kitchen dataset (Tenorth, Bandouch & Beetz, 2009), shown in Figure 2. 23, focuses on taking and dropping objects in a kitchen in both humanistic (realistic) and robotic ways. The data was recorded by four cameras in the top corners of the recording location. The video clips are RGB coloured, with a resolution of 384x288, and frame rate of 25 frames/second. The TUM Kitchen dataset focuses on ten low-level activities occurring in the kitchen: reaching, reaching up, taking something, lowering an object, releasing grasp, opening a door, closing a door, opening a drawer, closing a drawer, and carrying.

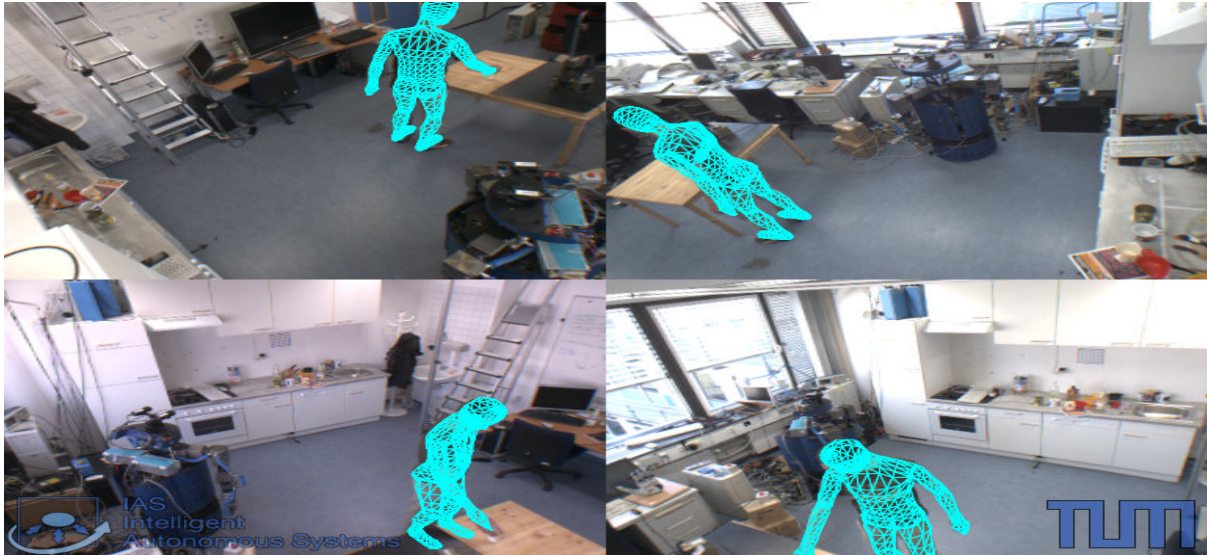


Figure 2. 23: The TUM Kitchen dataset.

The Collective Activity Dataset (Choi, Shahid & Savarese, 2011) is a five-activity dataset that includes crossing, waiting, queueing, walking, and talking. This dataset was recorded in an outdoor environment, and contains 44 video clips. The 10th frame in each clip is annotated with the image location of the person, activity ID, and pose direction.

The HMDB51 dataset (Kuehne, Jhuang, Garrote, Poggio & Serre, 2011) is a video dataset that contains 51 action classes and approximately 7,000 video clips. The creators of this dataset collected the majority of the data from films, and the rest from YouTube and from public databases, such as the Prelinger archive. The dataset is clustered into five main groups: general facial actions, facial actions with object manipulation, general body movements, body movements with object interaction, and body movements for human interaction.

The UCF101 is one of the most popular video datasets for human actions (Soomro, Zamir & Shah, 2012). It consists of data collected from YouTube and contains 101 action categories and 13,320 videos. It has a large variety of camera motions, object appearances and poses, object scales, viewpoints, cluttered backgrounds, and illumination conditions. The actions in the UCF101 action dataset are divided into five primary categories: sports, body-motion only, human–object interaction, playing musical instruments, and human–human interaction. A sample frame of six action classes is shown in Figure 2. 24.



Figure 2. 24: Six action classes from UCF101.

The MPII Cooking Activities dataset (Rohrbach, Amin, Andriluka & Schiele, 2012) is an activity dataset focused on cooking and preparing meals, presenting multiple people preparing different dishes in video clips of varying durations. The MPII was recorded using a single camera located at the top centre of the kitchen, as can be seen from Figure 2. 25. The total number of activities is 65. The MPII dataset focuses on low-level activities, such as chopping, adding, arranging, closing, and moving.

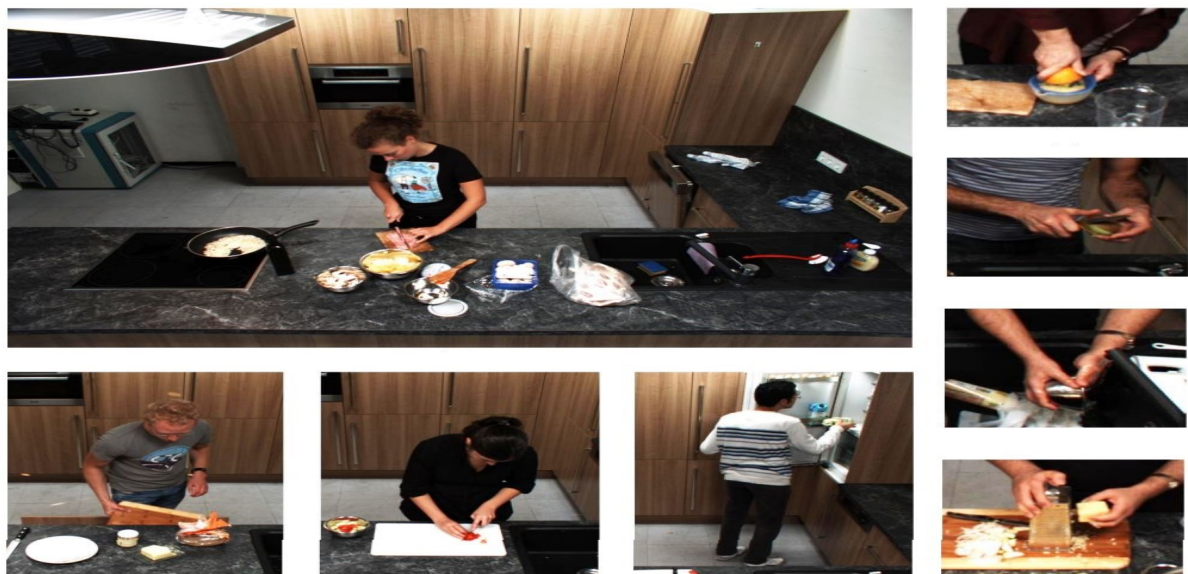


Figure 2. 25: Sample from the MPII Cooking Activities dataset.

A volleyball activity dataset was created by Waltner, Mauthner and Bischof (2014). The dataset contains six videos with seven challenging volleyball activities from real-life

professional matches in the Austrian Volley League. The seven activities are: serve, reception, setting, attack, block, stand, and defence/move.

The EPIC Kitchen dataset (Damen et al., 2018) focuses on the first-person vision, using a head-mounted camera and performing the activities in front of the camera, as shown in Figure 2. 26. The EPIC dataset is the largest first-person kitchen dataset, and was recorded in 32 different kitchens, with a resolution of 1280x720 pixels and 60 frames/second. This dataset focuses on object recognition and speech to identify the activities.



Figure 2. 26: The Epic Kitchen dataset.

2.6.2.2 Depth Camera Activity Datasets

The Cornell Activity Datasets, or the CADs, are two datasets produced by Cornell University. The first, called CAD-60 (Sung, Ponce, Selman & Saxena, 2012), which consists of 60 video clips. In the CAD-60 dataset, 12 activities are performed by four different subjects in front of an RGB-D camera. These activities are: rinsing the mouth, brushing teeth, wearing contact lens, talking on the phone, drinking water, opening a pill container, cooking (chopping), cooking (stirring), talking on the couch, relaxing on the couch, writing on the whiteboard, and working on a computer. In the CAD-60, five different locations were used to record the activities, and the RGB-D camera was located at one side of the room at a human body level, as shown in Figure 2. 27.



Figure 2. 27: The CAD-60 dataset.

The second Cornell dataset is the CAD-120 (Koppula, Gupta & Saxena, 2013). In this dataset, 120 video clips were recorded using an RGB-D camera located at a human body level on one side of the room. In the CAD-120, four subjects perform ten high-level activities. These activities are: making cereal, taking medicine, stacking objects, unstacking objects, microwaving food, picking up objects, cleaning objects, taking food, arranging objects, and having a meal. These high-level activities were performed in a working lab location, and the activities are not related to the location itself. An example is shown in Figure 2. 28. In both CAD datasets, the camera was located at body height, which makes it prone to occlusion. Also, the activities were staged, the observer can easily notice the acted activity in the majority of videos, and some of the activities were performed in a non-realistic place.



Figure 2. 28: The CAD-120 dataset.

The MSR DailyActivity 3D Dataset (Kurakin, Zhang & Liu, 2012) contains 16 different activities: drinking, eating, reading a book, making a phone call on a cell phone, writing on paper, using a laptop, using a vacuum cleaner, cheer up, sitting still, tossing paper, playing a game, laying down on a sofa, walking, playing the guitar, standing up, sitting down. All the activities were performed in the same place, as shown in Figure 2. 29, with two different postures: standing and sitting. The Kinect sensor was placed in a low-level location, the same level as the monitored person or lower, which is not a practical camera location.



Figure 2. 29: The 16 activities in the MSR DailyActivity dataset.

The G3D action recognition dataset (Bloom, Makris & Argyriou, 2012) was also recorded using a Kinect. This dataset contains the RGB information, depth information, and skeleton information. It also includes a range of gaming actions. The G3D dataset features 10 subjects performing 20 gaming actions: punch right, punch left, kick right, kick left, defend, golf swing, tennis swing forehand, tennis swing backhand, tennis serve, throw a bowling ball, aim and fire a gun, walk, run, jump, climb, crouch, steer a car, wave, flap and clap (Bloom, Makris & Argyriou, 2012).

The SBU Kinect interaction dataset is a two-person interaction dataset created by Yun et al. (2012). The interactions include approaching, departing, kicking, punching, pushing, hugging, shaking hands, and exchanging.

The Multiview 3D Event dataset (Wei, Zhao, Zheng & Zhu, 2013) is a large-scale dataset that was created using Kinect to record eight subjects performing eight activities: drink with a mug, call with a cell phone, read a book, use a mouse, type on a keyboard, fetch water from a dispenser, pour water from a kettle, and press button. All the activities were recorded in

almost the same location using three cameras located at head height. A sample from the Multiview 3D Event dataset is shown in Figure 2. 30.



Figure 2. 30: Multiview 3D Event dataset.

The Sphere dataset is a public dataset that is recorded and managed by a team from the University of Bristol as part of the Sphere project (Tao, Burghardt, Hannuna, Camplani, Paiement, Damen & Craddock, 2015). Various staged activities were performed and recorded by five different people in front of a camera. The dataset contains 13 action categories per clip: sit still, stand still, sitting down, standing up, walking, wiping the table, dusting, vacuuming, sweeping floor, cleaning a stain, picking up, squatting, and upper body stretching. As shown in Figure 2. 31, the camera was located at the top corner location, the most practical location. However, all 13 activities were performed in the same place.



Figure 2. 31: Sample of the Sphere-H130 dataset.

2.6.2.3 Fall Datasets

This section covers some of the most popular fall datasets, recorded with different camera models and in different locations.

Auvinet et al. (2010) presented a fall dataset using a multi-camera system. Eight cameras in top corner locations were used to record the falls, as shown in Figure 2. 32. In this dataset, 22 scenarios covering different types of fall were performed. The falls are noticeably staged (clear acting in the fall), and the locations of the staged fall are visible, as there are helping pads that were used to absorb the impact of the fall. The frame rate is 120 frame/second, and the resolution is 720x480.

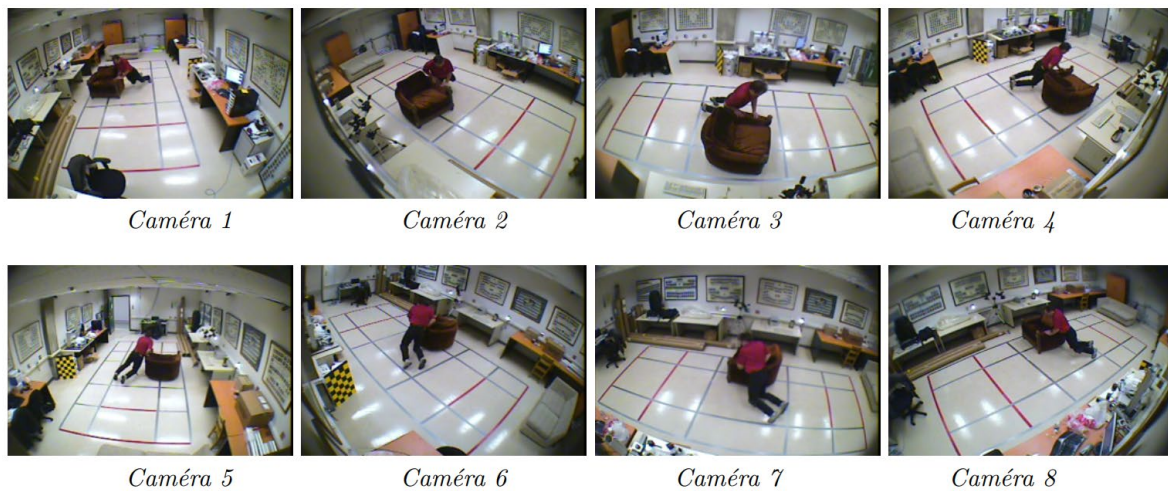


Figure 2. 32: Sample from the multiple cameras fall dataset.

Charfi et al. (2013) created a fall dataset that covers falls in six locations: office, lecture room, home 1, home 2, and coffee room 1, coffee room 2. A sample is shown in Figure 2. 33. The researchers used the top corners to record the falls. The frame rate is 25 frames/second, and the resolution is 320x240 pixels. The main issue with this dataset is the visible helping pad that was used to help to absorb the impact of the fall.



Figure 2. 33: Sample from the fall detection dataset.

The University of Rzeszow created the UR Fall Detection dataset (Kwolek & Kepski, 2014), which contains 30 falls and 40 ADL videos. Two Microsoft Kinects and accelerometers were used to create the dataset. One Kinect was located in the centre of the ceiling, and the second located on one side of the room at body height. A sample of the dataset is shown in Figure 2. 34.



Figure 2. 34: UR Fall Detection dataset.

2.6.3 Metrics

Different metrics have been used to assess the performance of the various methods. Some researchers have used accuracy as the only metric to evaluate their work. By contrast, others have used several different metrics, such as simplicity, specificity and accuracy, which helps to provide more details about the efficiency of their method.

Accuracy provides information about how many correct predictions the system produces and is the measurement of how close the model comes to identifying all the correct results, i.e. the percentage of the time the model is correct, as shown in Equation 2. 14. Accuracy is a prevalent performance metric, and is the ratio of correctly identified activities (Joint Committee for Guides in Metrology, 2008).

$$Accuracy = \frac{\text{Correctly identified class}}{\text{Total testing class}} \times 100$$

$$= \frac{TP + TN}{TP + TN + FP + FN} \times 100 \quad 2. 14$$

In the above equation, True Positive (TP) values are correctly identified positive values. True Negative (TN) values are correctly predicted negative values. A False Positive (FP) occurs when the class is negative, and the prediction shows as positive. A False Negative (FN) occurs when the actual class is positive, but it is predicted as negative (Pauly, Hogg, Fuentes & Peel, 2017). This is explained in the confusion matrix in Table 2. 2.

Table 2. 2: The confusion matrix explaining the TP, FP, FN, and TN.

Actual Class	Predicted class	
	Positive	Negative
	Positive	TP
Negative	FP	TN

Another popular metric used by researchers is precision (Positive Predictive Value). This measures the accurate predictions among the retrieved measurements; in other words, precision measures the exactness of a result (Joint Committee for Guides in Metrology, 2008).

It is calculated as the number of true positives divided by the number of true positives plus the number of false positives (Davis & Goadrich, 2006; Pauly et al., 2017). Precision is determined using Equation 2. 15.

$$Precision = \frac{TP}{TP + FP} \quad 2. 15$$

Recall (sensitivity, or true positive rate) is another popular performance metric, and measures the system's ability to identify all the relevant activities within a dataset. It is calculated as the number of true positives divided by the number of true positives plus the number of false negatives (Davis & Goadrich, 2006; Pauly et al., 2017). In other words, it measures the proportion of states (activities) that tested positive, as shown in Equation 2. 16.

$$Recall = \frac{TP}{TP + FN} \quad 2. 16$$

Specificity (true negative rate) is the opposite of sensitivity, and measures the proportion of states (activities) that tested negative, and it will show how good the methods are at avoiding false alarms as shown in Equation 2. 17.

$$Specificity = \frac{TN}{TN + FP} \quad 2. 17$$

The F1 score is the harmonic average of precision and recall, and indicates the system's performance stability. The formula for the F1 score is shown in Equation 2. 18.

$$F1 \text{ score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad 2. 18$$

Cohen Kappa is a statistical measurement metric that is used to measure the agreement between two raters. The formula for Cohen Kappa is shown in Equation 2. 19.

$$Kappa \text{ value} = \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e} \quad 2. 19$$

Where p_e is the hypothetical probability of chance agreement, and p_o is the relative observed agreement among raters, which is identical to accuracy.

Researchers have used many metrics in their work to analyse and measure performance. Chen, Fan and Cao (2015) used accuracy, precision, recall, and F1 score to measure the performance of their model; while others (Jalal, Kamal & Kim, 2015; Guesgen, 2015; Raeiszadeh & Tahayori, 2018) have used only accuracy as a validation unit in measuring the performance of activity identification by sensors and cameras.

Dubois and Charpillet (2013) used sensitivity and specificity to assess their model, for each activity. Other researchers (Cilla et al., 2009; Liu et al., 2014; Wu et al., 2014; Tsai et al., 2017) used accuracy with details from the confusion matrix to measure the performance of their models in identifying activities.

If the task is to identify uncommon activities (incidents) such as rare disease or fall, specificity must be used as it covers uncommon activities. A fall is an important activity that needs to be identified; it is better to have a system that detects a fall that did not happen than to have a system that is unable to detect a fall that does happen. For activity recognition that does not require uncommon activity to be identified, specificity as a metric is not required.

Chapter 3 Research Scope

In this chapter, an overview of the major problems in ADL monitoring from video data for assisted living is presented. After that, the problems selected to be addressed by this work are discussed.

There are a number of current problems for the assisted living systems based on video data that the majority of researchers are working on, as presented in Chapter 2:

- People detection
- People tracking
- Activity recognition
- System evaluation
- System usability:
 - Internet connectivity
 - Scalability and retrofitting
 - Affordability
 - Privacy
- Care response

The first step in an activity monitoring system is detecting the monitored people. Detecting and distinguishing a person and isolating them from their environment and surrounding objects can be a challenge. Detection of multiple people is an even bigger challenge, as sometimes these people are partly or fully occluded by objects or by each other.

After detecting the monitored people, they need to be tracked to identify the activity. The system must follow the recognised person continuously, even through occlusions, to identify their activities.

Tracking of multiple people is more difficult, as they need to be individually tagged and differentiated. Another issue is occlusion. If the detected people occlude each other or are occluded by objects, ensuring continuous tracking becomes a challenge.

Once the system detects and tracks the person, the next step is to identify the activity. A significant amount of work has been done in the activity recognition area, as mentioned in

Chapter 2. However, the prior work has been designed to identify only a limited number of activities, and used staged activities or staged environments. Moreover, the performance of the proposed approaches leaves space for improvement.

Some researchers focused on identifying posture-level activities such as standing, waving, or clapping. However, posture-level activities do not represent ADL. Therefore, other researchers addressed identifying low-level activities, such as using a computer mouse, picking an object, and chopping. However, the low-level activities give only a limited view of ADL that a person may be involved in. This has lead researchers to work on identifying high-level activities such as cooking, washing up, or working on a computer. To date, only a small number of high-level activities have been addressed, and these activities have often been implemented in scenarios that are different from real-life circumstances (e.g. staged activities). It is expected that the performance of these methods may deteriorate when tested on real-life scenarios. For instance, some of the studies that cover high-level activities were performed in a single location or in a small number of locations. This may make the resulting methods not practical, as they do not cover real-life environments.

Evaluation of the activity identification performance requires appropriate datasets. Creating realistic datasets with recorded ADLs and labelling those datasets for subsequent use in performance testing represents a separate challenge. The majority of the available datasets cover either posture-level activities or low-level activities, rather than high-level activities. They also cover a limited number of locations within a typical residential environment and often provide artificial viewing angles for the covered areas (e.g. a straight view from the chest height that would be difficult to achieve in a real-life environment).

In addition to improving the monitoring system performance, improving the system usability is another key challenge in assisted living. The available Internet bandwidth is one issue that affects the usability because a monitoring system usually needs to transfer data between the monitored residence and processing servers and career facilities. To improve the usability, the system needs to transfer the minimum amounts of data necessary to perform the job successfully. For example, it can be transferring data only when an incident occurs or is expected to occur, instead of continually sending data.

The system scalability and ease of retrofitting into existing dwellings also affect the system usability. Increasing the number of sensors that are used for activity recognition may not always be possible due to hardware limitations. Modifying a house to make it suitable for the installation of sensors in all the required areas can be expensive and time-consuming. To solve these problems, researchers suggested using different types of sensors that can be easily installed and require less modification for the dwelling such as image sensors (i.e., camera), as shown in the previous Chapter.

The system cost is another issue that affects the usability of an activity monitoring system. Activity monitoring systems need to be as cost-efficient as possible. It is desirable to have an affordable monitoring system that can be installed in a large number of domestic dwellings. Therefore, a possible focus of further research can be on using standard off-the-shelf sensors (e.g. standard RGB cameras) instead of any specialised equipment.

Privacy is a key concern for any activity monitoring system. Non-image sensors usually have better privacy compared to image sensors. However, image sensors can provide more information about an ongoing activity. Researchers suggested using wearable sensors to get more data, but there are many issues with wearable sensors, as presented in Chapter 2. These include the need to be in contact with the body all the time, which affects the system flexibility and usability. Image sensors do not interfere with a person's activities, but pose perhaps the most significant concern for invading privacy, making it one of the biggest challenges for video-based monitoring.

Care response is the final step for an activity monitoring system. The provided response will be based on the identified activities. The response can be a call to a family member, informing a caretaker, calling an ambulance, or providing information to a doctor. The system needs to be able to determine the most appropriate response depending on the circumstances.

A summary of the discussed research challenges is presented in Table 3. 1.

Table 3. 1 Potential research problems for a video-based assisted living system

	Improve Performance	Improve Usability	Introduce New Applications
Detection	Detect multiple persons.	Reduce the system cost and the required Internet bandwidth. Improve system privacy.	
Tracking	Track multiple persons.	Reduce the system cost and the required Internet bandwidth. Improve system privacy.	
Action Recognition	Recognise more pose-based and low-level activities.	Reduce the system cost and the required Internet bandwidth. Improve system privacy.	Recognise high-level ADL and IADL.

This work proposes to address the following three major gaps:

1. Create a novel method for identifying high-level activities using high-level features.

High-level activities will help in understanding the daily living routine for a detected person. Creating a system that can identify basic Activities of Daily Living (ADL) and Instrumental Activity of Daily Living (IADL) using high-level features is the first goal of this work. A secondary goal is to identify sudden activities, such as falls.

There are many reasons for focusing on high-level ADL and IADL. These reasons are psychological, economical, sociological, physical, and behavioural. Using the monitoring of high-level ADL and IADL, researchers can create solutions for many physical and mental problems that can affect people and help in providing a more comfortable and safe independent living environment.

2. Create a dataset for performance evaluation of high-level activity identification.

In order to assess how successfully the first goal is achieved, performance of any new methods will need to be evaluated. As pointed earlier in this Section, the majority of existing datasets used to test activity identification performance only cover posture-based or low-level activities. Therefore, a dataset is required to cover a range of ADL and IADL in realistic real-life scenarios. The dataset would also need to be labelled with high-level activities as well as high-level features used by the proposed methods.

3. Evaluate the effectiveness of different high-level features for activity identification.

Another main issue that is addressed in this work is studying the effectiveness of the high-level features, i.e. feature selection for activity identification. A central idea in this work is to use high-level features for activity identification. Location, time, posture, and orientation are selected as the high-level features that will be used as an input for the proposed algorithms. The usefulness of the features in identifying the activities will be evaluated by testing different feature combinations, including one feature (location), two features (location and time), three features (orientation or posture with location, and time) and all four features combined (posture, orientation, location, and time).

In addition to the three major gaps above, this work also addresses some of the usability issues, including affordability and the ease of retrofitting. It is proposed to use inexpensive IP cameras that can connect via Ethernet and Wi-Fi as the image sensors. Such cameras can provide a cost-effective solution helping with the affordability of the system. Using Wi-Fi cameras requires less modification to a house to install the system, as it only needs providing a power source.

There are several problems in the assisted living area that this work does not address. The reason for this is the limited time available. In addition to that, there are some other reasons for each one of the discussed problems, as presented below.

This work does not focus on the detecting and tracking problems, as this area has been covered extensively in the past and the existing approaches achieve very high performance in person detection and object recognition such COCO (Lin et al., 2014), and YOLO9000 (Redmon & Farhadi, 2017).

The available Internet bandwidth for such monitoring systems can be a significant problem. The leading Internet providers and network vendors such as BT Group, Ericsson and Nokia (Patzold, 2019) are targeting this issue. Internet service providers and mobile operators are providing more bandwidth and data transfer speed to customers following advances in wired and wireless networks technologies. Therefore, this issue is also outside the scope of this work.

Privacy is a significant issue for activity monitoring systems, as they collect in-depth information about people's personal lives. System design plays a major role in addressing

privacy concerns. Ideally, only the minimum necessary information should be communicated outside a person's home. This suggests that most of the processing should be done inside sensors or other system components installed inside a person's dwelling. In this work, the focus is on developing and evaluating novel processing algorithms. The hardware aspects are left outside the scope, as this is a very different subject area also requiring specialist hardware prototyping facilities. There are currently many ongoing efforts focusing on creating faster, smaller, and more power-efficient processors that support AI and parallel processing, and can be installed in a camera to perform all the necessary processing inside. This will also help with addressing privacy concerns for assisted living systems.

Chapter 4 Identifying High-Level Activities

In this work, the aim is to create a system that can identify high-level activities of daily living in an indoor environment using high-level features. The objective is to achieve high performance in identifying high-level activities and sudden activities.

This chapter is organised as follows. Section 4.1 presents a justification for the psychological and technical reasons that lead to the selection of high-level features. Section 4.2 explains the methods that were used to identify the activities based on two features. In Section 4.3, the methods that were used to identify the posture and orientation are presented. Finally, Section 4.4 shows how all the high-level features are combined to identify the activities, with a summary of the proposed approach.

4.1 Selecting Features for Activity Identification

Feature selection usually has a major influence on the performance of classification tasks.

The proposed model combines the following high-level features to identify the activities:

- Spatial: characterising the location of the person in the environment.
- Temporal: characterising timing of the activities.
- Posture: characterising the person's body posture.
- Orientation: characterising the person's orientation relative to objects of interest in the environment.

It is expected that combining these high-level features can improve the activity recognition performance because they can identify overlapped activities that share the same location but different posture and orientation, activities that share multiple locations, activities that can share multiple postures, and multiple orientations.

These high-level features are picked based on technical and nontechnical reasons, as explained in the following sections.

4.1.1 Nontechnical Justifications

Multiple factors can affect human activities in both indoor and outdoor environment (Paulus, 1983; Fiske, 1992; Triandis, 1994; Anderson & Bushman, 2002; Glanz, Rimer & Viswanath, 2008; Burleson, Lozano, Ravishankar, Lee & Mahoney, 2018). These include:

- Environment and location
 - Surrounding environment
 - Person's location in the environment
 - Available space for the person in the environment
- Health
 - Mental health, memory, and attention span
 - Physical ability
- Group influence on individual future activities and activities sequence
- Time
 - Duration (the duration of past activities may affect the future activities sequence and duration)
 - Time of the day, week, month, and year
- Cultural, language, and social norms
- The type of clothes that the person wears
- Lifestyle
- Age
- Religion
- Gender
- Marital status
- Personality and communication skills

Location and the surrounding environment have a big impact on the activity, as some activities are normally performed in a single location such as cooking (kitchen) or in a few locations such as sleeping (bed or sofa). The time factor will determine the sequences of activities. If the person spends a long time on a single activity, it may lead to less time spent or cancellation for the following activities. Time of the year will determine the type of seasonal activities and outdoor activities such as ice skating. Physical ability will determine the person ability to perform the activities and the type of activities. It is also affected by the person pose and stamina.

For the proposed system, the assumption is that some factors are not essential in determining the activities, because they will be the same or very similar among all users of the system. Also, it is hard to identify some factors from images, and our system will depend only on

camera feeds. Such factors include different cultural backgrounds, language, age groups, marital status, and religion. In this study, the focus is on identifying human activities and not identifying the person identity. For that reason, some of the nontechnical factors are not considered including the clothing, gender, and some of the social life factors. In addition to that, this work targets the indoor activities. Therefore, the time of month, and the year is not considered.

Also, it is an essential assumption in this study that all the people that the system will monitor can perform the basic activities of daily living and do not need a carer to help them to perform their ADL and IADL. In addition to that, the monitored people are presumed to have no mental problems that could affect their everyday behaviour and the performance of the monitoring system.

The focus for this work is mainly on the features that can be observed from the cameras and will help in identifying human activities. For that reason, the environment and the person location are selected as the key features to provide information about the person's activities. The person's location has a significant influence on the activities. For example, activities in the kitchen are significantly different from activities in the office. The second key feature is the activity duration and the sequence of activities. They will help in predicting future activities based on the current and past activities. Physical ability is another main factor that can be observed from videos and images. It can help to differentiate between activities and cover several factors like moving, standing and falling. For that reason, the posture and orientation are selected.

4.1.2 Technical Justifications

For this work, the person activities are identified from images. The camera can be used to detect a person's location. The location of a person or the spatial feature can give information about the person's current activity. For instance, if the person is in front of a cooker, then they may be cooking. However, relying on the location alone may not be sufficient to identify the activity correctly as can be seen from Table 4. 1.

Table 4. 1: Likely problems in using location only for activity recognition (AL: Actual Location, DL: Detected Location, AA: Actual Activity, Ln: Location, An: Activity, Tn: Time)

Time	T1	T2	T3		T1	T2	T3		T1	T2	T3
AL	L1	L2	L2		L2	L2	L2		L2	L3	L3
DL	L1	L1	L2		L2	L4	L2		L2	L2	L3
AA	A1	A2	A2		A2	A2	A2		A2	A3	A3
	Problem 1				Problem 2				Problem 3		
AL	L1	L1	L2		L2	L4	L2		L2	L2	L3
DL	L1	L1	L2		L2	L2	L2		L2	L2	L3
AA	A1	A2	A2		A2	A2	A2		A2	A3	A3
	Problem 4				Problem 5				Problem 6		
AL	L1	L2	L2						L2	L3	L3
DL	L1	L2	L2						L2	L3	L3
AA	A1	A1	A2						A2	A2	A3
	Problem 7								Problem 8		

Table 4. 1 gives several examples when the current location does not correspond to the current activity. Problems 2 and 5 in Table 4. 1 correspond to the cases of an ongoing activity, when the location is detected incorrectly for a brief time, or the actual location changes for a short time, for instance, when the person steps away and then quickly comes back to what they were doing.

Problems 1 and 3 in Table 4. 1 are the instances when the location is detected incorrectly at the start or the end of an activity. Problems 4 and 6 in Table 4. 1 demonstrate the situations where a new activity has started, but the person is still at a previous location, for instance, when the two locations partially overlap. Similarly, problems 7 and 8 in the table show the cases when the location has changed, but the previous activity is continuing.

Using temporal features can help in addressing some of these problems. In particular, the system can monitor how long the person has spent at a particular location and use that information in combination with the location data to recognise the activity. The intuition here is that if a person has spent enough time at the location, then they are more likely to be engaged in the corresponding activity, rather than just passing by. This temporal information can help in solving Problems 2 and 5 in Table 4. 1.

Another concern is when two different activities can occur at the same location or when the same activity can happen at multiple locations. Location and time alone cannot solve these problems. Therefore, more features are required to improve performance.

Using the person's orientation and posture can further improve the performance of activity recognition, particularly when changing from one activity to another. This is likely to help address Problems 4, 6, 7, and 8 in Table 4. 1. If the detected person spends some time standing next to the cooker and facing the cooker, this is more likely to mean that the activity is cooking. In this example, four features are considered: spatial (location next to the cooker), temporal (spending some time in this location), posture (standing), and orientation (facing the cooker). By combining these features, it is expected that the system performance in identifying the activity will increase.

This method can also be used to detect a sudden fall by combining pose and spatial information. Falling is one of the biggest health hazards that can affect people at home, especially older adults (Scuffham, Chaplin & Legood, 2003). Pose features will help in identifying sudden activities, but it should be used with location to differentiate between daily activities and sudden activities. For example, lying in the kitchen is likely a fall, but lying on the bed is likely to mean sleeping.

Based on the above considerations, it is expected that combining spatial, temporal, posture, and orientation features will improve the detection accuracy for identifying activities. The four selected features will be extracted from images. Other features would require different sensors. While they may improve the performance, they will also increase the cost and complexity of the system. This will affect the system affordability and availability, which are some of the issues that this work aims to address as is explained in Chapter 3.

4.2 Spatial and Temporal Features

First, this work investigates how spatial and temporal features (i.e. the location and the activity duration at the location) can be used to identify activities of daily living in an indoor environment.

4.2.1 Identifying Location

To find the location of a person in a room, two algorithms are proposed:

- Convolutional Pose Machine (CPM) (Wei et al., 2016).

- Background Subtraction with K-nearest neighbours (BS-KNN) (Barnich & Van Droogenbroeck, 2010; Yin, Wang, Li, Liu & Zhang, 2015).

The CPM is used to detect people in the room in the first instance. If the CPM does not detect any person in the video, then BS-KNN is used. All the training and testing will be conducted on videos that contain a person in the videos, as the focus of this work is activity identification and not human detection. Therefore, any videos without a person will not be used to build the training model and in testing the work and methods. This is important for algorithms like BS-KNN, which otherwise may not be able to distinguish a person and other moving objects. All the objects in the rooms are manually labelled by defining rectangular areas corresponding to each object as can be seen in Figure 4. 1. If the person's head coordinates are in the region for a given labelled object, then that the person is labelled as located at that object. If the person is in an unlabelled region, then the location is unlabelled. It usually corresponds to the case for the walking activity. It may also correspond to a sudden activity as can be seen in Figure 4. 1.



Figure 4. 1: Shows the hand labelling for the objects in the locations

The CPM provides fourteen points for the detected human body parts, as explained later in this Chapter. Head coordinates are used to represent the person's location. The head coordinates from the CPM output as shown in Table 4. 2 are taken and compared to the manually labelled locations of objects in the room as presented in Figure 4. 2 and Figure 4. 5. Based on this comparison, the location for the detected person can be found. Full details about the CPM are explained later in Section 4.4.

In cases where the CPM does not produce an output due to not detecting a person, the BS-KNN or the previous state for the CPM can be used. The method with the highest total

performance will be selected based on the experimental evaluation. More details are presented in Chapter 6. The BS-KNN is one of the most popular techniques that isolate and detect moving objects in videos by isolating the stationary background in input images. The BS-KNN, however, will detect any movement in a video, including small moving objects like pets and changes in lighting. To avoid the detection of small objects and changes in lighting, a contour size threshold is introduced. Only moving objects with the contour size above the threshold are detected as a person. The threshold is determined experimentally to achieve the best human detection results for the used datasets. The erode and dilate morphology operators are used (Rössl, Kobbelt & Seidel, 2000) to further eliminate undesirable detection. Finally, the detection contour size is adjusted based on the video resolution. The full details are presented in Chapter 6.

When using BS-KNN, the centre of the top edge of the detected contour for the moving object is selected to represent the person's location. This approximates using the head coordinates for the CPM, as shown in Figure 4. 2.

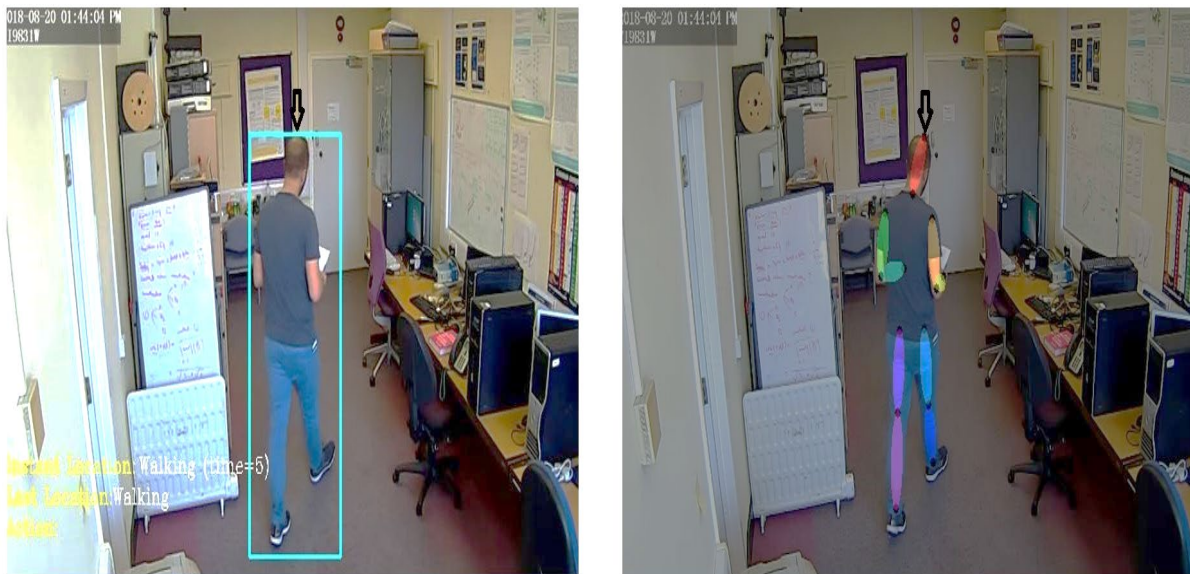


Figure 4. 2: Represents the location detection using the BS-KNN on the left side, and the CPM on the right side, the arrow represents the selected point for the BS-KNN and the CPM.

4.2.2 Applying Time Thresholds

A simple way to use the temporal features to differentiate between engaging in the activity at a particular location and just passing by is using a time threshold, as shown in Figure 4. 3. A new activity is detected only when the person spends longer than the time threshold at a new location. This can help to address issues 2 and 5 in Table 4. 1. For example, if the person

is cooking and then goes to the dining table to bring something, the detected location may change, but the actual activity of cooking will continue. Adding a time threshold for changing the activity following a change in location will help in such cases, as it will not change the activity immediately after the new location is detected. However, this can introduce new errors when switching from one activity to another. Introducing a time threshold will cause delays when detecting a new activity. It may also result in missing short activities if their duration falls below the threshold. Therefore, selecting the threshold value can have a significant effect on the system performance (Al-Wattar et al., 2016).

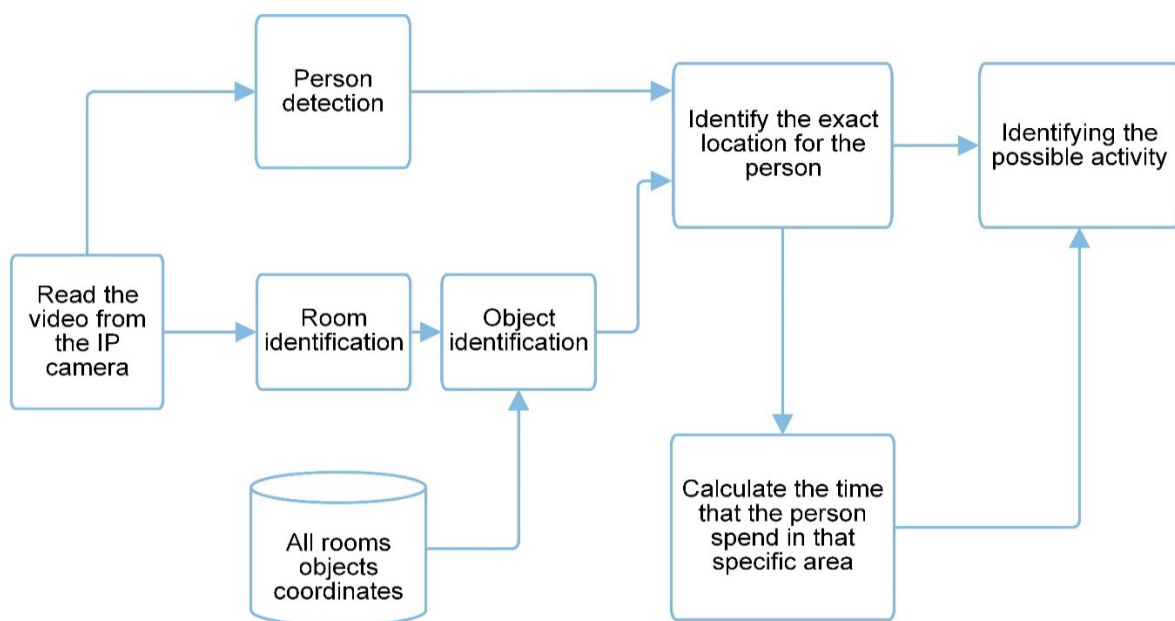


Figure 4. 3 The system design based on spatial and temporal features

The goal would be to achieve a balance between setting too low threshold resulting in detecting spurious activities, and too high threshold making the system less responsive and missing short activities. This suggests that an activity-specific threshold can be better than an activity-independent one.

Activities in the kitchen (like cooking) may tend to be shorter, i.e. requiring a lower time threshold than activities in the bedroom (like sleeping). Infrequent activities, such as using a washing machine, may need a higher threshold, as it is more likely that the person is just passing by a washing machine without the intention to use it. Activities that normally take longer durations, such as cooking, may similarly require a higher threshold compared to activities that are usually short, such as making a cup of tea.

To calculate activity-specific thresholds, it is proposed to analyse available video data to identify the instances when the location changes for a limited period, but the activity continues (Problems 2 and 5 in Table 4. 1). The time threshold for each activity can then be calculated as the minimum, maximum, or average duration of such instances.

To identify an activity, the first step is finding the person in the video. Once the detected person is in a defined object location, a counter is started. When the counter reaches a specific threshold, the activity changes to the activity assigned to this particular location. Changing from one activity to another depends on the previously labelled locations. If the person moves from point A to point B and spends the time that is greater than the threshold the activity will change to activity B as the new activity, and restart counting until the person moves to a different location. It then restarts the counter and counts again. The system output is shown in Figure 4. 5. A pseudocode of the steps for the fixed time threshold algorithm is presented in Figure 4. 4.

The Fixed Time Threshold Algorithm

Inputs:

Time threshold, T
 Activities assigned to each location, $A(location)$
 Sequence of observed locations, $L_i, i = 0..t$

Start:

Initialise $current_activity = A(L_0)$

Initialise $count = 0$

for $i = 1..t$

if $L_i = L_{i-1}$

$count = count + 1$

if ($count > T$):

$current_activity = A(L_i)$

output $current_activity$

end if

else:

$count = 1$

end if

end for

end

Figure 4. 4: The pseudocode for the fixed time threshold algorithm

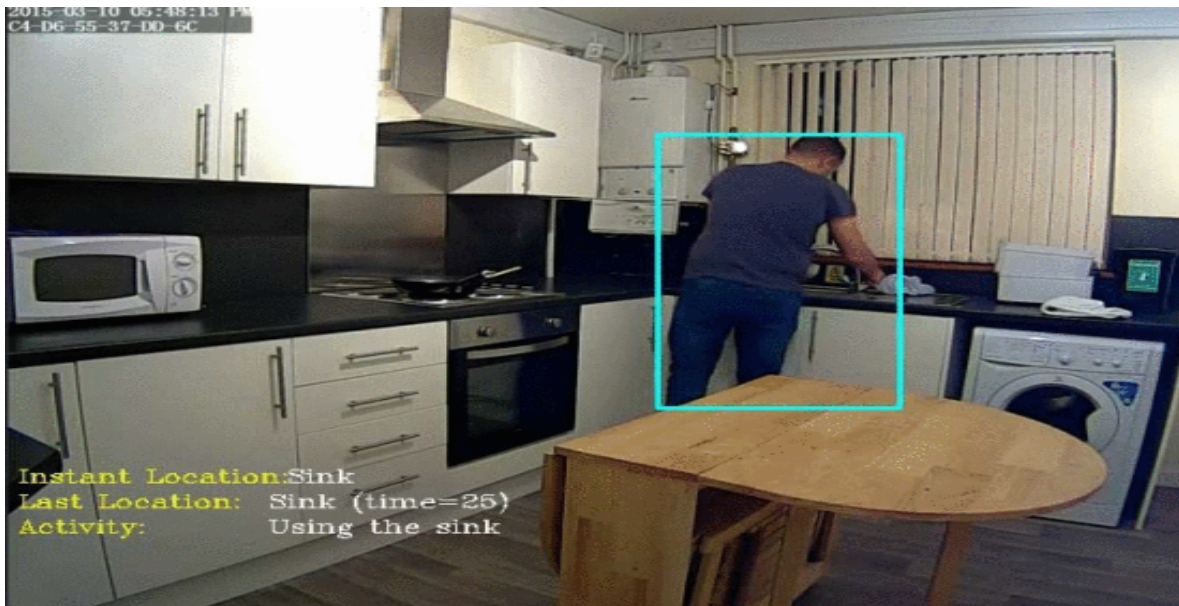


Figure 4. 5 The activity based on the fixed threshold output

The steps for the activity-specific thresholds algorithm are presented in Figure 4. 6.

The Adaptive Time Threshold Algorithm

Inputs:

Activity specific threshold $Adaptive_threshold(activity)$ for each activity,
 Activities assigned to each location, $A(location)$
 Sequence of observed locations, $L_i, i = 0..t$

Starts:

```

Initialise  $current\_activity = A(L_0)$ 
Initialise  $count = 0$ 
for  $i = 1..t$ 
  if  $L_i = L_{i-1}$ 
     $count = count + 1$ 
    if ( $count > Adaptive\_threshold (Activity (L_i))$ ):
       $current\_activity = A(L_i)$ 
      output  $current\_activity$ 
    end if
  else:
     $count = 1$ 
  end if
end for
end

```

Figure 4. 6: The pseudocode for the adaptive time threshold algorithm

Activity specific thresholds are determined in three different ways as the minimum, the maximum, and the average duration for a given activity. These values are calculated from the training data. Full details about the results for each way are presented in Chapter 6.

4.2.3 Applying Hidden Markov Models

An alternative way for combining spatial and temporal features is using Hidden Markov Models (HMM). HMM have been used by many researchers working on activity and voice recognition, as shown in Chapter 2. The process of transitioning from one activity to another activity in time can be naturally mapped onto the process of transitioning between states in an HMM. Using such mapping will require accepting the Markovian assumption that the next state in an HMM is only dependent on the current state and not past states. When applied to this work, it would mean that the next activity will depend only on the current activity. Strictly speaking, this may not always be the case in real life. However, it is expected the uncertainty about the future due to possible dependencies on the past could be approximated by the probabilistic nature of the HMM behaviour.

When using HMM, activities can be mapped onto the hidden states, while observed person's locations naturally map onto the observations generated in HMM. Time can be represented as the number of transitions between two given states. Continued activity is equivalent to an HMM transitioning from a state to the same state. Using this approach, HMM essentially combines the spatial and temporal features in one model.

Hidden Markov Models can be used to detect activities by estimating the likelihood of hidden states at a given moment in time from a sequence of generated observations. In this work, the sequence of observations is the sequence of observed locations for the person. The overall design for using HMM and the two proposed features (location and duration) to identify activities is presented in Figure 4. 7.

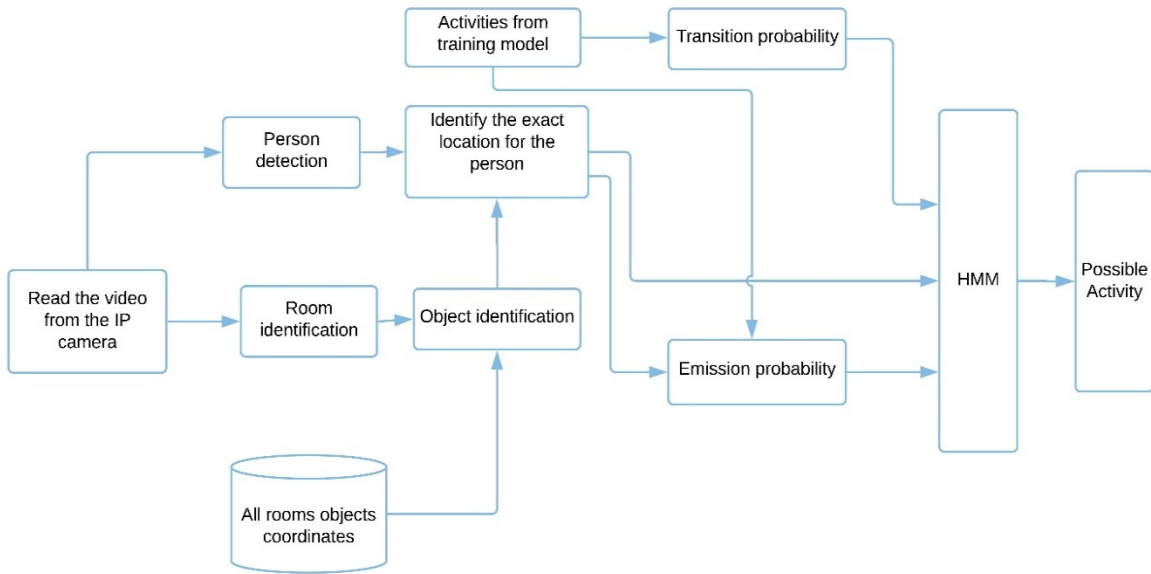


Figure 4. 7 Block diagram for HMM with spatial and temporal features

For any HMM, the emission and transition probabilities need to be calculated from training data in order to build the model (Russell & Norvig, 2016). To calculate the transition probabilities matrix, it is necessary to calculate the probability of changing from one activity to another (Rabiner, 1989):

$$\begin{aligned}
 \text{Transition Probability} &= P(A_t|A_{t-1}) \\
 &= \frac{P(A_t, A_{t-1})}{P(A_{t-1})} \\
 &\approx \frac{N(A_t, A_{t-1})}{N(A_{t-1})},
 \end{aligned} \tag{4.1}$$

where A_t represents the current activity at time t , A_{t-1} represents the previous activity, and t represent the time step that is associated with the transition, P represent the probability and $N()$ is the number of times the given activity or transition appears in the training data.

The emission (observation) probability is also calculated from the same training data (Rabiner, 1989) as the probability that the location at the current time t is L_t , given that the current activity is A_t :

$$\begin{aligned}
 \text{Emission Probability} &= P(L_t|A_t) \\
 &= \frac{P(L_t, A_t)}{P(A_t)}
 \end{aligned}$$

$$\approx \frac{N(L_t, A_t)}{N(A_t)} \quad 4.2$$

where A_t represents the current activity at time t , and L_t represents the location at time t . $N()$ is the number of times the given activity or the combination of activity and location appear in the training data.

The objective is to predict the hidden state A_t at time t , given the sequence $L_{0:t}$ of observed locations up to time t . There are several algorithms that can be used to calculate the probability of a hidden state given the history of past observations in HMM. These algorithms are presented in the following Sections.

4.2.3.1 The Forward Algorithm

The Forward algorithm can predict the hidden state at a specific time, given the history of observation for the previous states (Russell & Norvig, 2016). The Forward algorithm is based on the following equation:

$$\alpha_t(A_t) = P(L_t|A_t) \sum_{A_{t-1}} P(A_t|A_{t-1}) \alpha_{t-1}(A_{t-1}), \quad 4.3$$

where α_t is the estimate for the probability of $(A_t, L_{0:t})$ at the current time t , α_{t-1} represents the previous probability estimate at time $t - 1$, $P(A_t|A_{t-1})$ is the transition probability, and the $P(L_t|A_t)$ is the emission probability for the HMM. α_t is also referred to as the forward probability. The activity at time t can be predicted as the most likely activity according to the forward probabilities:

$$\text{Possible Activity} = \text{Argmax}_{A_t} (\alpha_t(A_t)) \quad 4.4$$

The HMM model can be built from the labelled training videos by calculating the emission and transition probabilities for each activity using equations 4. 1 and 4. 2.

The initial probabilities are assigned by assuming that all the activities in any location start with walking:

$$\begin{aligned} \alpha_1(A_1) &= 1 \text{ for } A_1 = W, \\ \alpha_1(A_1) &= 0 \text{ for } A_1 \neq W, \end{aligned} \quad 4.5$$

where W represents the “walking” activity. After that, the forward probability values are calculated iteratively based on equation 4. 3 for each activity until step t is reached. The

activity with the largest forward probability will be the expected activity at that time as shown in equation 4. 4. The steps for the Forward algorithm are as presented in Figure 4. 8:

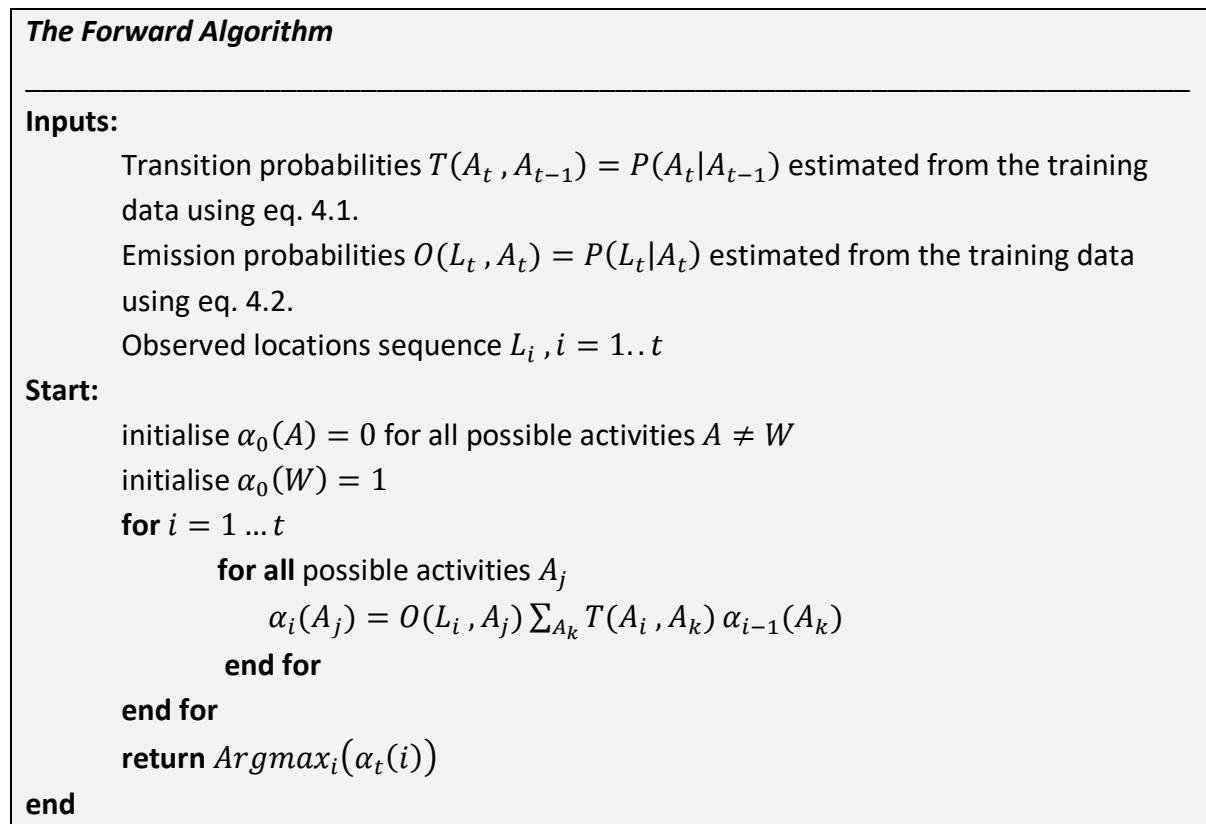


Figure 4. 8: The pseudocode for the Forward algorithm

4.2.3.2 The Forward-Backward Algorithm

The Forward-Backward algorithm is the Forward algorithm with backward smoothing. It can be used to improve estimates for the likelihood of past states using the subsequent observations. To compute the Forward-Backward algorithm, the forward and backward probabilities need to be calculated for each activity. The probability of an activity given the full observation sequence is then calculated as the product of the forward and backward probabilities. The activity with the maximum probability will represent the expected activity (Rabiner, 1989; Russell & Norvig, 2016).

In order to build the HMM model, the emission and transition probabilities need to be produced. These probabilities can be calculated from the training data, as explained earlier, using equations 4. 1 and 4. 2. Then forward probabilities α_t will need to be calculated based on the observation sequence until the current time t . The forward probabilities in equation 4. 3 can be presented in a matrix form as follows (Russell & Norvig, 2016):

$$\alpha_{0:t}^{\circ} = C_t^{-1} \alpha_{0:t-1}^{\circ} T O_{L_t} , \quad 4.6$$

where

- $\alpha_{0:t}^{\circ}$ is the vector of the forward probabilities for all states (activities) based on the observation sequence up to time t .
- C_t^{-1} is the scaling factor introduced to ensure that all probabilities in $\alpha_{0:t}^{\circ}$ add up to 1.
- $\alpha_{0:t-1}^{\circ}$ is the vector of the forward probabilities for all states based on the observation sequence up to time $t-1$.
- T is the transition probability matrix where the column index represents the target state and the row index represents the start state.
- O_{L_t} is the diagonal matrix with the values representing the probabilities of observing L_t for each state (activity).

The backward probability uses the observations from t until the end of the available observation sequence at time $N > t$. It is the probability that the process will generate that remaining observation sequence $L_{t:N}$ starting at time $t-1$ and from the given A_{t-1} state. The backward probabilities can also be represented in the matrix form (Russell & Norvig, 2016):

$$B_{t-1:N}^{\circ} = C_t^{-1} T O_{L_t} B_{t:N}^{\circ} , \quad 4.7$$

where

- $B_{t-1:N}^{\circ}$ is the vector of backward probabilities for all states (activities) A based on the observations from time t until the end of the available observation sequence at time N , $P(L_{t:N} | A_{t-1} = A)$.
- $B_{t:N}^{\circ}$ is the vector of backward probabilities for all states based on the observations from time $t+1$ until N .
- The remaining symbols are the same as in equation 4.6.

To calculate the backward probabilities $B_{t:N}^{\circ}$, the initial values for $B_{N:N}^{\circ}$ are set to 1 for all states (activities) since the initial state is assumed as given.

The forward and backward probabilities can be combined using the Bayes rule and the conditional independence of $L_{0:t}$ and $L_{t+1:N}$ given A_t in order to calculate probabilities for all states (activities) at time $t < N$ given the observation sequence $L_{0:N}$ as follows:

$$Y_t(i) = P(A_t = i | L_{0:N}) = \alpha_{0:t}^\circ(i) B_{t:N}^\circ(i) \quad 4.8$$

After that, the activity i with the maximum value $Y_t(i)$ is predicted as the possible activity at time t using the Forward-Backward algorithm:

$$\text{Possible Activity} = \text{Argmax}_i (Y_t(i)) \quad 4.9$$

The steps for the Forward-Backward algorithm are as follows:

The Forward-Backward Algorithm

Inputs:

Transition probabilities $T(A_t, A_{t-1}) = P(A_t | A_{t-1})$ estimated from the training data using eq. 4.1.

Emission probabilities $O(L_t, A_t) = P(L_t | A_t)$ estimated from the training data using eq. 4.2.

Observed locations sequence $L_i, i = 1..N$

Analysed activity time $t < N$

Start:

initialise $\alpha_0(A) = 0$ for all possible activities $A \neq \text{Walking}$

initialise $\alpha_0(\text{Walking}) = 1$

initialise $B_N(A) = 1$ for all possible activities A

for $i = 1 .. t$

for all possible activities A_j

$$\alpha_i(A_j) = O(L_i, A_j) \sum_{A_k} T(A_i, A_k) \alpha_{i-1}(A_k)$$

end for

for all possible activities A_j

$$\text{Normalise } \alpha_i(A_j) = \frac{\alpha_i(A_j)}{\sum_{A_k} \alpha_i(A_k)}$$

end for

end for

for $i = N-1 .. t$

for all possible activities A_j

$$B_i(A_j) = \sum_{A_k} B_{i+1}(A_k) O(L_{i+1}, A_k) T(A_i, A_k)$$

end for

for all possible activities A_j

$$\text{Normalise } B_i(A_j) = \frac{B_i(A_j)}{\sum_{A_k} B_i(A_k)}$$

end for

end for

```

for all possible activities  $A_j$ 
     $Y_t(A_j) = \alpha_t(A_j)B_t(A_j)$ 
end for
return  $Argmax_i(Y_t(i))$ 
end

```

Figure 4. 9: The pseudocode for the Forward-Backward algorithm

The Forward-Backward algorithm cannot be used to predict activities in real-time because it relies on knowing observations beyond the current point (i.e. future observations) for performing the backward probabilities calculations. Therefore, it is not useful for real-time assistance relying on real-time activity detection. However, it can be used to improve predictions for past activities. This can be used to improve the analysis of long term behaviour trends. For example, it can be applied at the end of the day or end of the week to improve analysis of past activities for any trends detection.

4.3 Posture and Orientation Features

Multiple methods to detect human pose are presented in Chapter 2, such as the pictorial structure (Felzenszwalb & Huttenlocher, 2005; Ferrari, Marin-Jimenez & Zisserman, 2009; Andriluka, Roth & Schiele, 2009), hierarchical models (Tian, Zitnick & Narasimhan, 2012), sequential prediction (Cao et al., 2016), the convolutional architecture (Tome, Russell & Agapito, 2017), and the Convolutional pose machine (CPM) (Wei et al., 2016). It is proposed to use the CPM for extracting posture and orientation features, as explained in the following sections.

4.3.1 Convolutional Pose Machine

The CPM is a prediction framework that learns the spatial model using the convolutional neural network. The CPM is an articulated pose estimator that can predict the human body parts from multiple convolutional networks that work on belief maps. The CPM consists of a series of convolutional networks that work to produce belief maps for each part, as shown in Figure 4. 10. The human body parts that the CPM are trained to find are fourteen parts. Once the CPM detects the fourteen parts, it creates the full pose for the detected body from the belief map, with no need for a clear model style for the detected body pose.

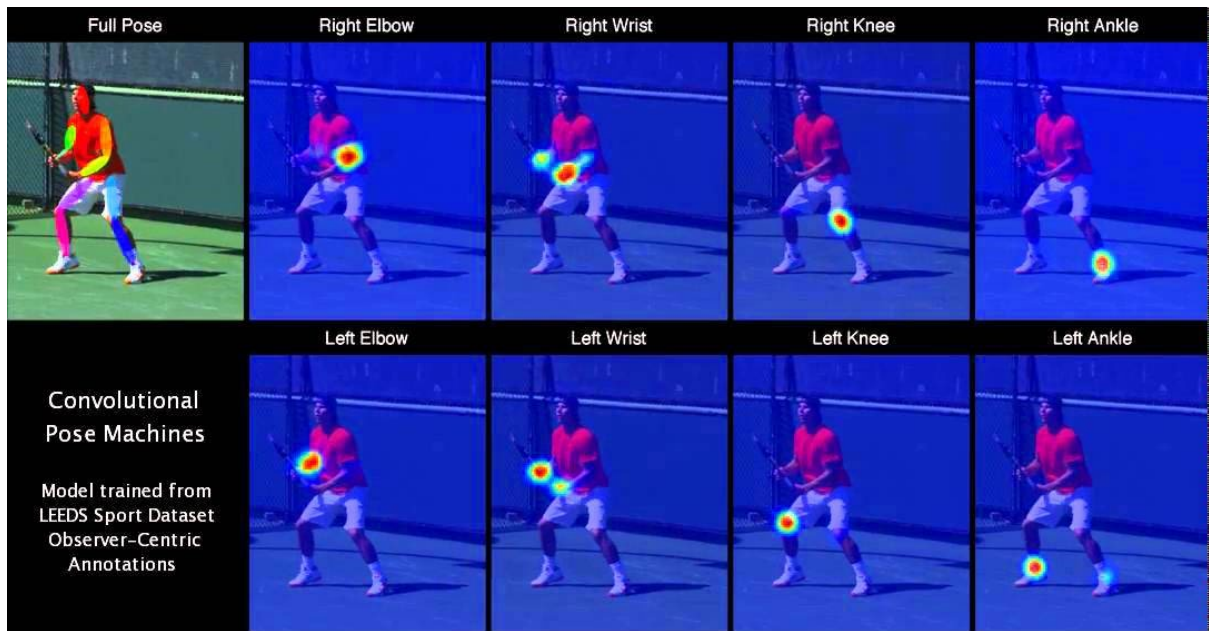


Figure 4. 10 The identifies body parts by the CPM (Wei et al., 2016)

The main reasons for selecting the CPM are that it is a very accurate method for detecting the human pose and it has the ability to detect a partly occluded body by predicting the partly occluded body parts with a reasonable level of accuracy. Also, it has comparatively modest computational requirements (Wei, Ramakrishna, Kanade & Sheikh, 2016), making it very cost-effective. The CPM achieved state-of-the-art performance on the MPII Human Pose dataset, as it delivered an average 8.4 % higher performance than its closest competitor. Also, the CPM attained a state-of-the-art performance when tested on Leeds Sports Pose (LSP) dataset, and FLIC dataset on detecting body parts outperforming all the competitive method.

The suggested CPM model has been trained to identify fourteen human body parts (Duckworth, Alomari, Charles, Hogg & Cohn, 2017). These parts are head, neck, right shoulder, right elbow, right hand, left shoulder, left elbow, left hand, right hip, right knee, right foot, left hip, left knee, and left foot. These fourteen points represent the human body, as shown in Figure 4. 11 Figure 4. 12 and Table 4. 2. The selected CPM can predict different human positions, and it can also detect partly occluded people, as shown in Figure 4. 11.



Figure 4. 11 The detection for a trained CPM

Table 4. 2 represents the detected X and Y coordinates for the person's body parts in the image. To identify the posture and orientation for the detected person, these X and Y values are post-processed so that they can be used to classify posture and orientation in the proposed design.

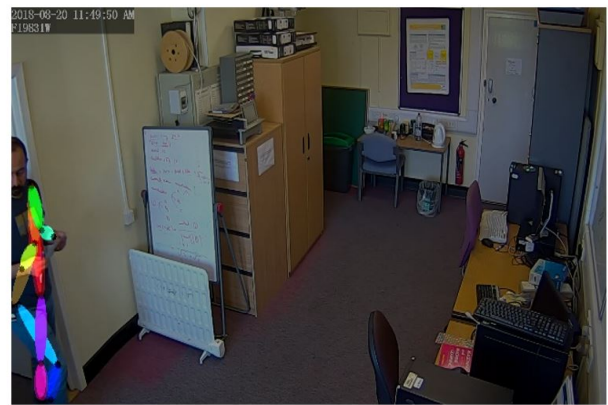
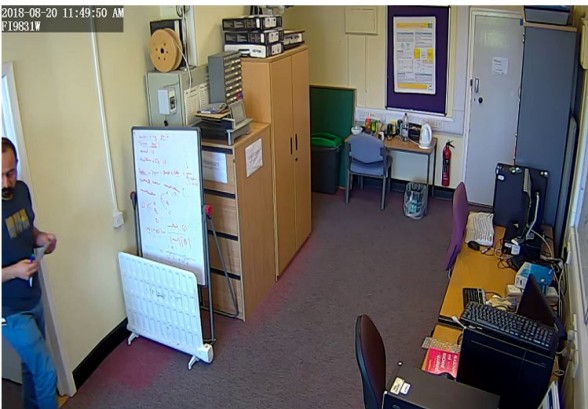


Figure 4. 12 The right image shows a partially occluded person, and the image on the left shows how the CPM manage to identify the human body parts (with some errors)

Table 4. 2 An example of the coordinated values for the fourteen points from the CPM output

	Joints	X coordinates	Y coordinates
1	head	30	267
2	neck	21	163
3	right_shoulder	21	167
4	right_elbow	27	218
5	right_hand	11	242
6	left_shoulder	21	164
7	left_elbow	31	205
8	left_hand	46	212
9	right_hip	8	275
10	right_knee	51	331
11	right_foot	43	363
12	left_hip	33	267
13	left_knee	32	330
14	left_foot	31	363

4.3.2 Posture and Orientation Classification

Both posture and orientation are determined by the positions of body parts relative to each other. Thus, using the absolute X and Y coordinates of body parts as inputs to a classifier will be inefficient. The absolute X and Y coordinates need to be normalised first before they can be used as inputs to a classifier. The normalised coordinates can then be used to create input feature vectors for posture and orientation classification. Two different methods for feature representation are proposed as described in the following sections.

4.3.2.1 Normalisation

The body part coordinates are normalised relative to the person's head location. Normalising the data will help in reducing dependencies on the person's location in a room, the distance from the camera, and the height of the person.

The first step in normalisation is to find the head and consider the head coordinates as the reference point for all the other coordinates. The head coordinates are set to be (0,0). The coordinates of the other body parts are expressed relative to the head coordinates. Finally, each of the fourteen X and Y values are divided by the maximum absolute X and Y coordinate values respectively to scale all coordinates to values between -1 and 1. The normalisation steps are presented in Figure 4. 13.

Normalisation

Inputs:

X_i and Y_i coordinates from the CPM output, presented in Table 4.2

Head coordinates X_{head} & Y_{head}

Start:

for $i = 1..14$

$$X_i = X_i - X_{head}$$

$$Y_i = Y_i - Y_{head}$$

end for

$$X_{max} = \max |X_i|$$

$$Y_{max} = \max |Y_i|$$

for $i=1 .. 14$

$$X_i = \frac{X_i}{X_{max}}$$

$$Y_i = \frac{Y_i}{Y_{max}}$$

end for

end

Figure 4. 13: The pseudocode for the normalisation process

The new normalised values for X_i & Y_i are shown in Table 4. 3.

Table 4. 3 the normalised values for the fourteen coordinates from the CPM output for Table 4. 2

	Normalised values	
	X coordinates	Y coordinates
1	0	0
2	-0.40909	-1
3	-0.40909	-0.96154
4	-0.13636	-0.47115
5	-0.86364	-0.24038
6	-0.40909	-0.99038
7	0.045455	-0.59615
8	0.727273	-0.52885
9	-1	0.076923
10	0.954545	0.615385
11	0.590909	0.923077
12	0.136364	0
13	0.090909	0.605769
14	0.045455	0.923077

4.3.2.2 Classification with Normalised Coordinates

The first approach to identify the posture and orientation is to feed all the normalised values from the CPM into a classifier, as shown in Figure 4. 14. Then, the trained classifier will be used to identify the posture and orientation.

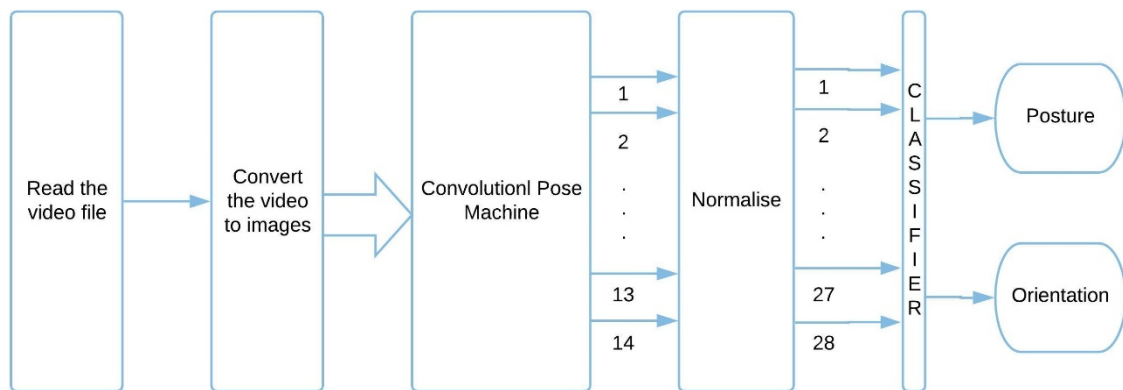


Figure 4. 14: The completed design for the normalisation with the classification method for identifying the posture and orientation algorithm

The feature vector will consist of 28 attributes corresponding to the 14 X and Y pairs. These features will be used with different classifiers to determine the best one for the task based on the achieved performance. More details about this are presented in Chapter 6.

The data are labelled with four possible postures and five possible orientations for the detected person. The postures are standing, sitting, falling (lying) and bowing (bending).

The possible orientations consist of facing the object, away from the object, facing left to the object, facing right to the object, and facing the floor. The orientation will be based on the torso (the upper body) for the detected person.

Labelling the orientation for the detected person can be performed relative to the object or relative to the camera. In this work, the orientation from the camera's perspective and not from the object's perspective is considered. For example, if the person is facing right in the image from the camera, the orientation will be facing right, irrespective of what it is relative to the object. The reason for that is to reduce the amount for hand-labelling. If the orientation is considered based on the object viewpoint, the detected person will have multiple orientations for the same location if this location shares multiple activities, and this will add more complexity.

Considering labelling based on the object perspective will have one advantage that one can use the same model for all the cameras, as facing the object will be same for all the objects. However, the video labelling will be very time consuming, as the videos will need to be labelled based on each object (location).

A summary of the classification approach with normalised coordinates is presented below:

- Use the normalisation algorithm to normalise the CPM output and produce the 28 features for all the data instances.
- Train the model on the training data to build classifiers to identify the detected person posture and orientation.
- Test the model on the testing data to check the performance of the classifiers for identifying the posture and orientation of the detected person.

4.3.2.3 Classification with Pairwise Distances

The second approach to identifying the posture and orientation is to calculate the Euclidian distances between pairs of different body parts to express their location relative to each other. It is expected that the pairwise distance may give more information about some activities, such as holding hands and carrying using both hands, shown in Figure 4. 15. For example, if the person is carrying something using both hands, the distance between the hands will be the same in any location, but the coordinates will depend on the location from the camera. This makes the pairwise features more useful for this type of activities. Also, identifying fall may be easier, as the distance between the hands and knees, the hands and feet, and elbows and knees will be shorter than during sitting and standing, as shown in Figure 4. 15.



Figure 4. 15: Activities that can be easily identified by the Pairwise method

The normalised coordinates from the CPM are used to calculate the Euclidian distances between each pair of the 14 body parts:

$$Distance(i,j) = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} \quad 4. 10$$

Where i and j are the two body parts, and X_j and Y_j are the corresponding coordinates. As the CPM produces 14 points, the number of pairwise distances will be 91.

Therefore, the input feature vector for a classifier will contain 91 attributes. Different classifiers will be tested, and the best one will be selected based on the accuracy. More details are presented in Chapter 6. The complete design for classification using pairwise distances to identify the posture and orientation is presented in Figure 4. 16.

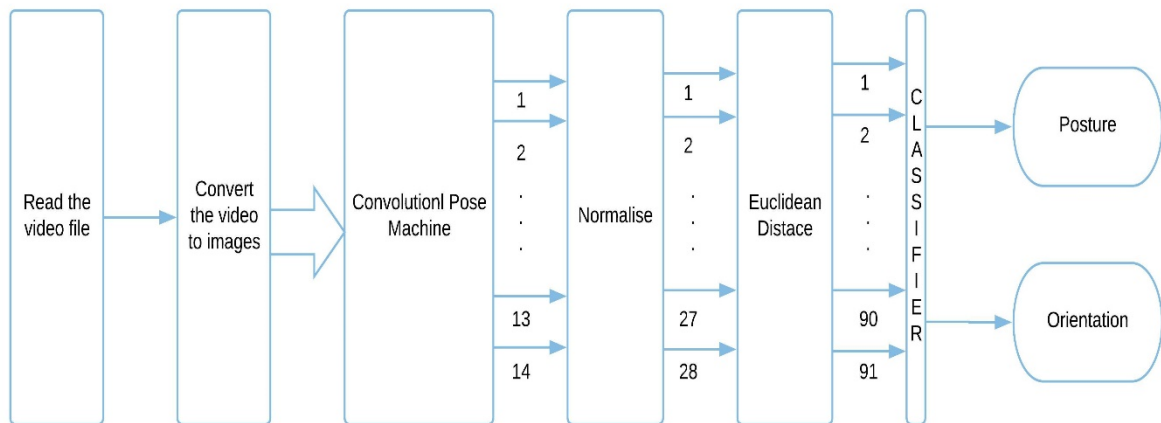


Figure 4. 16: Posture and orientation classification using pairwise distances.

A summary of the classification approach with pairwise distances is presented below:

- Normalise the CPM output using the normalisation algorithm and produce the 28 body parts coordinates for all the data instances.
- Produce the feature vectors for all the data instances as the Euclidean pair-wise distances between all body parts. The total number of features will be 91.
- Train the model on training data to build classifiers on the 91 features to identify the detected person's posture and orientation.
- Test the model on testing data to check the performance of the classifiers for identifying the posture and orientation of the detected person.

4.3.2.4 Classifiers

For both methods, the same classifiers with the same configuration will be used. The selected five classifiers are the Naïve-Bayes, the support Vector Machine, the Decision Tree, the Random Forest, and the Feedforward Neural Network classifiers.

A brief introduction for the selected five classifiers is presented. The used classifiers are:

• Naïve-Bayes Classifier

The Naïve-Bayes Classifier is a Bayesian classification algorithm which is based on applying the Bayes formula with naïve feature independency. The Naïve-Bayes classifier is a probabilistic supervised classifier. Although it is one of the simplest classifiers, it achieves high results with the high dimensionality of input data (McCallum & Nigam, 1998).

- The Support Vector Machine

The Support Vector Machine (SVM) is one of the most popular supervised learning classifiers. The SVM is a classifier that separates data based on hyperplanes, from the given training data, and categorises them based on hyperplanes depending on the number of classes (Suykens & Vandewalle, 1999).

- Decision Tree Classifier

The Decision Tree classifier is a supervised classifier that splits the data into smaller subsets and builds a decision tree at the same time. The Decision Tree classifier can be a graphical representation depending on certain conditions (Safavian & Landgrebe, 1991).

- The Random Forest Classifier

The Random Forest is also a supervised classifier that works by building many decision trees for producing the classes, and it is one of the most popular classifiers (Ho, 1995, Barandiaran, 1998).

- The Feedforward Neural Network

The Feedforward Neural Network is a multilayer perceptron (MLP) that consists of at least three layers, an input layer, a hidden layer, and an output layer. The Feedforward classifier is a supervised machine learning classifier. The MLP utilises the backpropagation for training. Each of the neurons in the hidden and output layers uses the activation function that map the weight for each neuron (Cybenko, 1989), full details about the Feedforward NN are in Chapter 2.

4.4 Combining Spatio-Temporal and Pose-Based Features

The proposed method for combining the high-level features will use HMM with the Forward algorithm, the Forward-Backward algorithm, and the Long Short-Term Memory.

4.4.1 Hidden Markov Models

The spatial features will be calculated, as explained previously in Section 4.2.1. The posture and orientation will be detected, as presented in Section 4.3. To combine the four features (location, duration, posture, and orientation), the HMM model will have hidden states mapped onto activities, while observations will be mapped onto the possible combinations of

location, posture, and orientation. Durations, as previously, will correspond to the number of transitions between two hidden states.

Emission probabilities need to be estimated from training data to build this new HMM. Instead of estimating the probability of observing a given location in a given hidden state (activity), it is now necessary to estimate the probability of observing a given combination of location, posture, and orientation for the given activity:

$$\begin{aligned}
 \text{Emission probability} &= P((P, O, L)_i | A_j) \\
 &= \frac{P((P, O, L)_i, A_j)}{P(A_j)} \\
 &\approx \frac{N((P, O, L)_i, A_j)}{N(A_j)}
 \end{aligned}
 \tag{4.11}$$

Where $(P, O, L)_i$ is a combination of posture, orientation, and location, A_j is the activity, and $N(x)$ is the number of times (count) the given observation combination or activity x occurs in the training data.

A possible problem with this approach is that a particular combination of posture, orientation, and location may not appear in training data due to the large number of different combinations and the limited amount of training data. In such cases $P((P, O, L)_i | A_j) = 0$, which may affect the accuracy of the resulting model. A different approach will be used to address this issue.

A similar issue has been addressed by other researchers, for instance, in-text prediction (Bikel, Schwartz & Weischedel, 1999). They used an efficient approach to address this problem by splitting the training data (vocabularies data) into two parts based on the frequency of the words, such as high-frequency and low-frequency words. Frequent words were labelled according to their actual appearance in the training data. Low frequency and never mentioned words were labelled based on the prefix and suffix such as labelling number like "90" as "twoDigitNum", words written in all uppercase letters like "BBN" as "allCaps", and "other" for punctuation symbols like ",", ". In this work, it is not possible to use the same approach as different features are used, and additional information, such as morphological structure, is not available. Therefore, a different approach is proposed.

Using the rules of conditional probability, the emission probability for a given combination of posture, orientation, and location (P, O, L) and a given activity A can be expressed as follows:

$$\begin{aligned}
 P(P, O, L|A) &= \frac{P(P,O,L,A)}{P(A)} \\
 &= P(P|O, L, A) \times \frac{P(O, L, A)}{P(A)} \\
 &= P(P|O, L, A) \times P(O|L, A) \times \frac{P(L, A)}{P(A)} \\
 &= P(P|O, L, A) \times P(O|L, A) \times P(L|A)
 \end{aligned}
 \tag{4. 12}$$

The following three assumptions are made:

- Assumption 1: Orientation and location are conditionally independent given activity.
- Assumption 2: Orientation and posture are conditionally independent given location and activity.
- Assumption 3: Posture and location are conditionally independent given activity.

Under Assumption 1, O and L are conditionally independent given A . That is:

$$P(O, L|A) = P(O|A) \times P(L|A) \tag{4. 13}$$

Substituting this into equation 4. 14 below gives the following:

$$\begin{aligned}
 P(O|L, A) &= \frac{P(O,L,A)}{P(L,A)} \\
 &= P(O, L|A) \times \frac{P(A)}{P(L,A)} \\
 &= P(O|A) \times \frac{P(L|A)}{P(L|A)} \\
 &= P(O|A)
 \end{aligned}
 \tag{4. 14}$$

The first term in the final product in equation 4. 12 can be expressed as follows:

$$\begin{aligned}
 P(P|O, L, A) &= \frac{P(P,O,L,A)}{P(O,L,A)} \\
 &= P(O, P|L, A) \times \frac{P(L,A)}{P(O,L,A)}
 \end{aligned}$$

$$= \frac{P(O, P|L, A)}{P(O|A)} \quad 4. 15$$

Under Assumption 2, O and P are conditionally independent given the joint event L, A . That is:

$$P(O, P|L, A) = P(O|L, A) \times P(P|L, A) \quad 4. 16$$

Substituting this into equation 4. 15, results in the following:

$$\begin{aligned} P(P|O, L, A) &= P(O|A) \times \frac{P(P|L, A)}{P(O|A)} \\ &= P(P|L, A) \end{aligned} \quad 4. 17$$

Finally, under Assumption 3, P and L are conditionally independent given A . Similarly, to the case of O and L , it can be shown that $P(P|L, A) = P(P|A)$. Substituting this result into equation 4. 17 gives in the following:

$$P(P|O, L, A) = P(P|A) \quad 4. 18$$

Substituting equations 4. 14 and 4. 18 into 4. 12 results in the following:

$$P(P, O, L|A) = P(P|A) \times P(O|A) \times P(L|A) \quad 4. 19$$

That is, the emission probability for a combination of three features (position, orientation, location) and a given activity can be expressed as a product of emission probabilities for the individual features for this activity. This equation is used to calculate emission probabilities when a particular combination of features does not occur in the training data (i.e. $P((P, O, L)_i|A_j) = 0$).

The proposed assumptions essentially state that the observed features (location, orientation, and postures) are primarily determined by the activity, and are independent of each other. In real life, posture and orientation may, in some cases, be depended on location. For instance, if the same activity can take place at different locations, the posture and orientation may be different at different locations and, thus, not independent. Therefore, it is useful to know how well the proposed assumptions are matched in the actual data to assess the accuracy of the proposed model and the potential effect of these assumptions on the model performance.

In order to evaluate the assumptions, Pearson's correlation coefficient can be used to test for independence. Pearson's correlation coefficient (r) has been used by many researchers (McCain, 1990; Ahlgren, Jarneving & Rousseau, 2003; Sedgwick, 2012).

Pearson's correlation coefficient is a measurement for the statistical relationship between two variables, i.e. the similarity between data pairs (White & McCain, 1998). The values for (r) vary between -1 to 1. When the correlation is positive, this means both values are correlated and moving in the same direction. When (r) is negative, it means negative linear correlation. The closer the values are to 1 or -1, the stronger is linear correlation. The formula for the Pearson correlation coefficient is as follows:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad 4. 20$$

Where r is the Pearson coefficient, n is the number of pairs in the provided data, x_i & y_i are the values for paired variables. \bar{x} & \bar{y} are the mean values for the x & y samples.

For the proposed assumptions, the following equations should hold in case of conditional independence, as is shown earlier:

- For assumption 1:

$$P(O|L, A) = P(O|A) ;$$

- For assumption 2:

$$P(P|O, L, A) = P(P|L, A) ;$$

- For assumption 3:

$$P(P|L, A) = P(P|A) ;$$

where P, O, L, A are posture, orientation, location, and activity, respectively. Pearson correlation will be used to test the strength of the relationships expressed by these equations to assess how suitable the proposed assumptions are for our data. The details and the results are presented in Chapter 6.

Once all the zero values for the joint emission probabilities $P((P, O, L)_i | A_j)$ are replaced with the product of emission probabilities for individual features, as shown in equation 4. 19, the joint emission probability values will need to be normalised to make the sum for all emission probabilities for a given activity to be equal to one:

$$\text{Normalised emission probability} = \frac{P((P, O, L)_i | A_j)}{\sum_k P((P, O, L)_k | A_j)} \quad 4. 21$$

After normalising, the final emission probabilities will be used with the transition probability equation 4. 1 in the resulting HMM model.

4.4.2 Long Short-Term Memory

Another approach that can be used instead of HMM is the Long Short-Term Memory (LSTM). Many researchers have used LSTM for time series prediction for activity recognition and text prediction (Graves et al., 2008; Ordóñez & Roggen, 2016; Guan & Plötz, 2017; Ma, Chen, Kira & AlRegib, 2019). Some researchers combine LSTM networks with CNN models to predict the activity (Baccouche, Mamalet, Wolf, Garcia & Baskurt, 2011; Guan & Plötz, 2017), as they use the CNN to extract the features and the LSTM to identify the activity from the extract features.

LSTM is a neural network with feedback connections and memory that can deal with long dependencies which help in remembering sequences of input information. That is why researchers use it for sequential data. LSTM can take a sequence of inputs and return a sequence of outputs or a single output. This can be used to predict activities based on sequences of observations from the cameras, which can include posture, orientation, and location. The memory in the LSTM will remember and consider past information while predicting future values. This can account for temporal features.

The LSTM input will be a sequence of the observed postures, orientations, and locations at each time step, as shown in Figure 4. 17. Each cell has a state that carries the relative information about the data sequence. The propagation of information between different time steps and between input and output is controlled via the cell gates. This allows the network to learn the relevant information and overcome the vanishing gradient problem, as explained in Chapter 2.

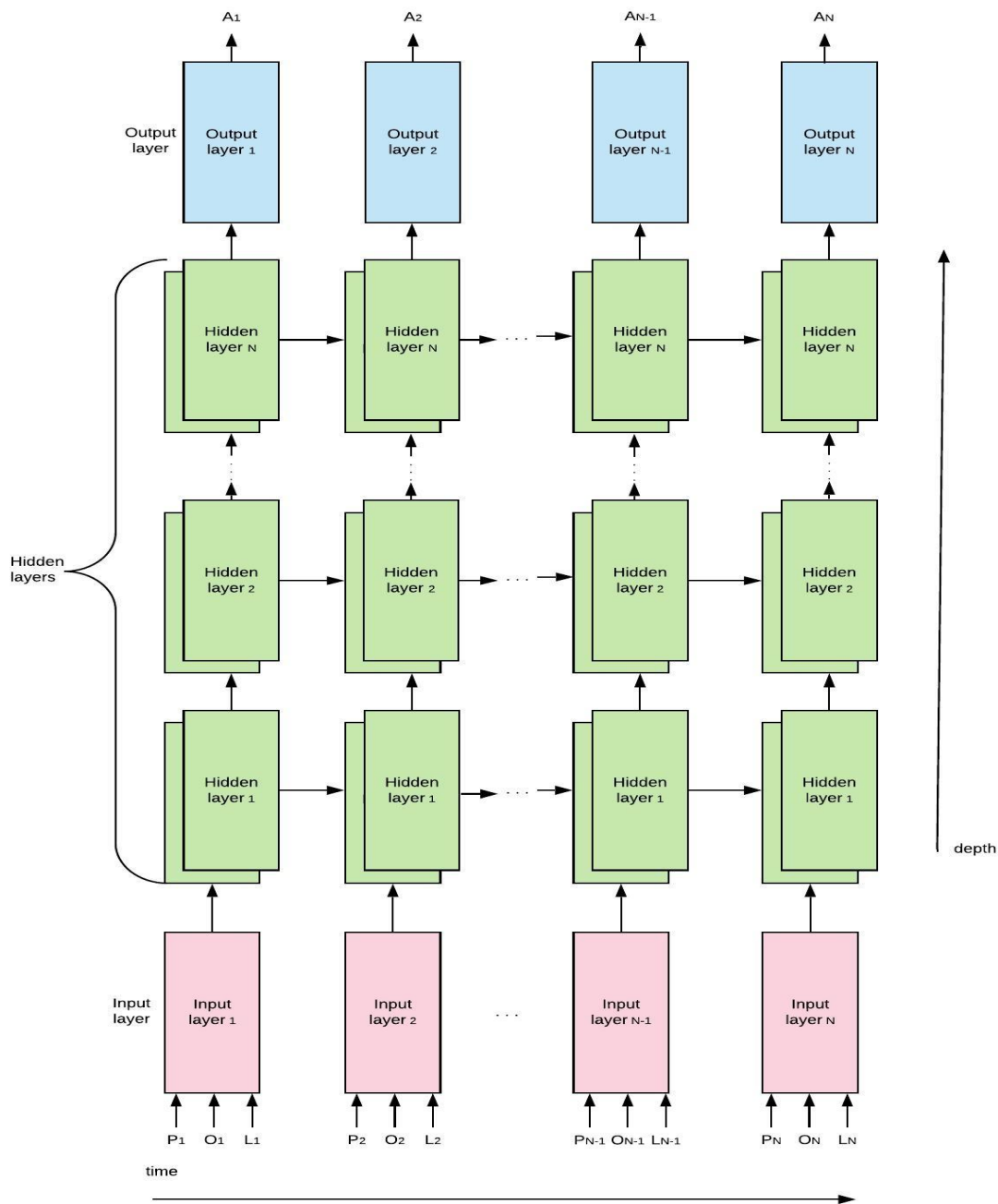


Figure 4. 17: LSTM network design

The LSTM network has three inputs corresponding to posture, orientation, location; and one output corresponding to the predicted activity. At each time step, the network produces the activity output based on the current posture, location, orientation inputs; and the current states of LSTM cells in each hidden layer, calculated at the previous time step. At each time step, new LSTM cell states are calculated to be used at the next time step. This is shown in Figure 4. 18. The values for the LSTM cell inputs and output are calculated; as explained in Chapter 2.

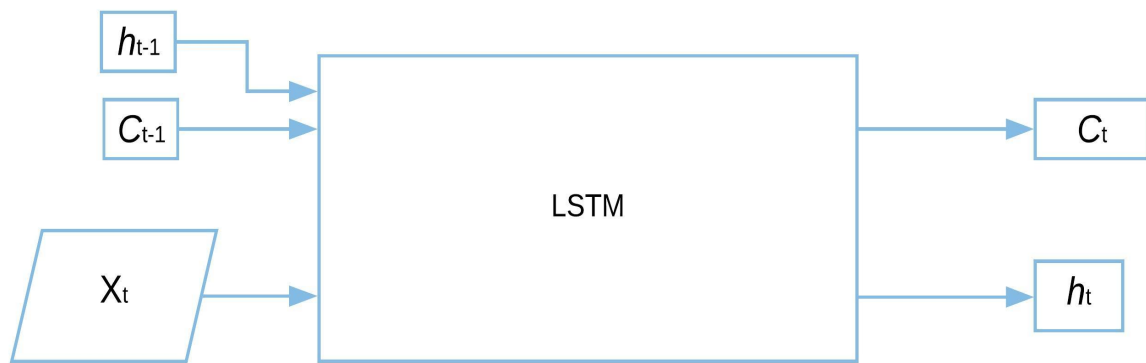


Figure 4. 18: The input and the output for the LSTM unit where h_t is the current hidden state, h_{t-1} is the previous hidden state, C_t is the cell state, and C_{t-1} is the previous cell state. X_t is the input vector.

The input layer consists of three nodes passing the input values of posture, orientation, and location to the first hidden layer. The output layer has the number of nodes equal to the number of possible activities. The outputs of each node are the probabilities for activities. The *softmax* function is used for the output nodes in the network.

The number of hidden layers will vary, and the number of hidden layers will depend on the LSTM model performance. Some researchers have shown that the deeper the NN the better the performance (Goodfellow et al., 2016). However, LSTM does not require very deep network like CNN to achieve good performance.

It is recommended to use two hidden layers for simple models as a start, and adjust the number based on performance. However, time-series prediction may take an extra layer or two (Heaton, 2015). For example, the work for these researchers used less than four layers (Baccouche et al., 2011; Wu, Jiang, Wang, Ye, Xue & Wang, 2015; Ibrahim, Muralidharan, Deng, Vahdat & Mori, 2016; Zhu et al., 2016; Shu, Todorovic & Zhu, 2017). Therefore, the number for the hidden layers in the proposed network will vary between one and five.

The number of LSTM cells in each hidden layer will also vary. The final configuration will be selected based on the model performance. There are no fixed rules to select the numbers of cells (nodes) for each hidden layer. Using too many nodes does not necessarily mean better performance for the network. Using too many nodes in the hidden layers may cause

overfitting. In addition to that, too many nodes will increase the amount of training time (Heaton, 2015; Ma et al., 2019).

As a starting point, Heaton (Heaton, 2015) suggested three methods to select the number of cells in the hidden layers:

- The number of cells in the hidden layers should be between the sizes of the input layer and the output layer.
- The number of cells is equal to 70% of the size of the input layer plus the size of the output layer.
- Start with a number that is less than twice the size of the input layer.

These methods will be considered as a starting point and then adjusted to select the configuration that achieves that best performance.

Network optimisation is an essential part of the work. Many methods can be used for optimisation and determine the values for the weights, biases, and initial values, such as Adagrad (Duchi, Hazan & Singer, 2011) and ADAM (Kingma & Ba, 2014). The method that achieves the best performance in less number of iteration will be used.

There is no fixed rule to select the batch size and the number of epochs. They need to be selected based on model performance (Heaton, 2015; Brownlee, 2018). Therefore, different models will be configured, and the one that achieves the best performance will be used.

The input sequence length (window size) that achieves the best performance will be selected, as there is no fixed rule to select it and it mainly depends on the dataset (Kudo, Toyama & Shimbo, 1999; Baccouche et al., 2011; Karpathy, 2015; Heaton, 2015; Brownlee, 2018). The sequence length and how it created with the results are presented in Chapter 6.

4.5 Complete Design

The overall framework of the proposed algorithms for combining the spatio-temporal and pose-based features are presented in Figure 4. 20 and Figure 4. 21. The HMM and the LSTM are used to predict the activities. The observed states will be the combinations of posture, orientation and location for the detected person. The hidden state will be the activity at a given time, as shown in Figure 4. 19.

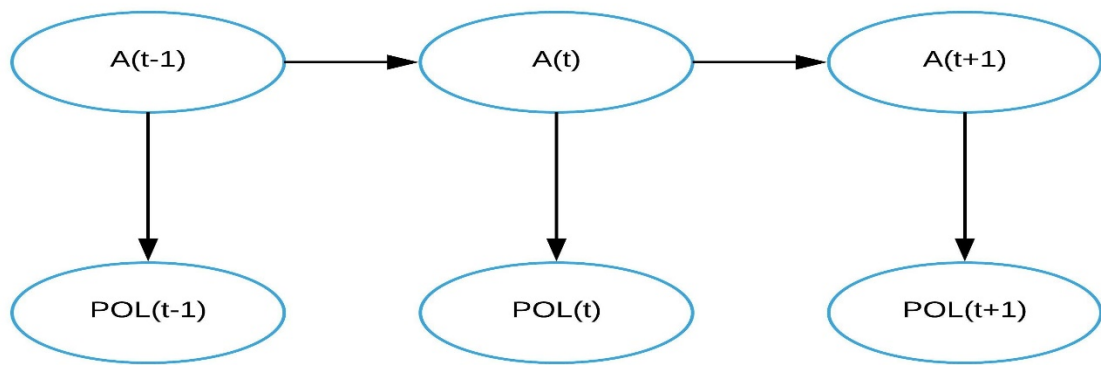
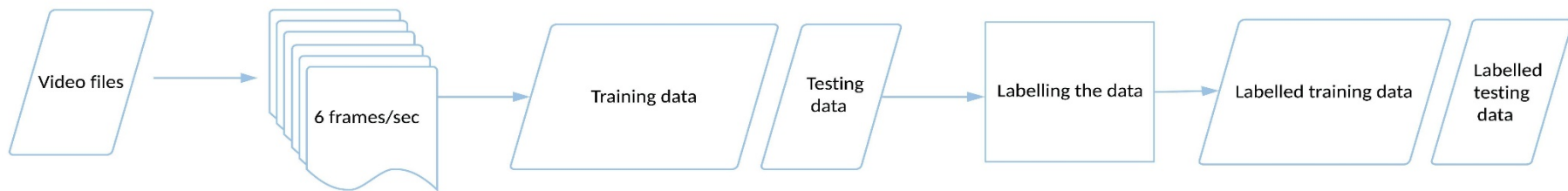
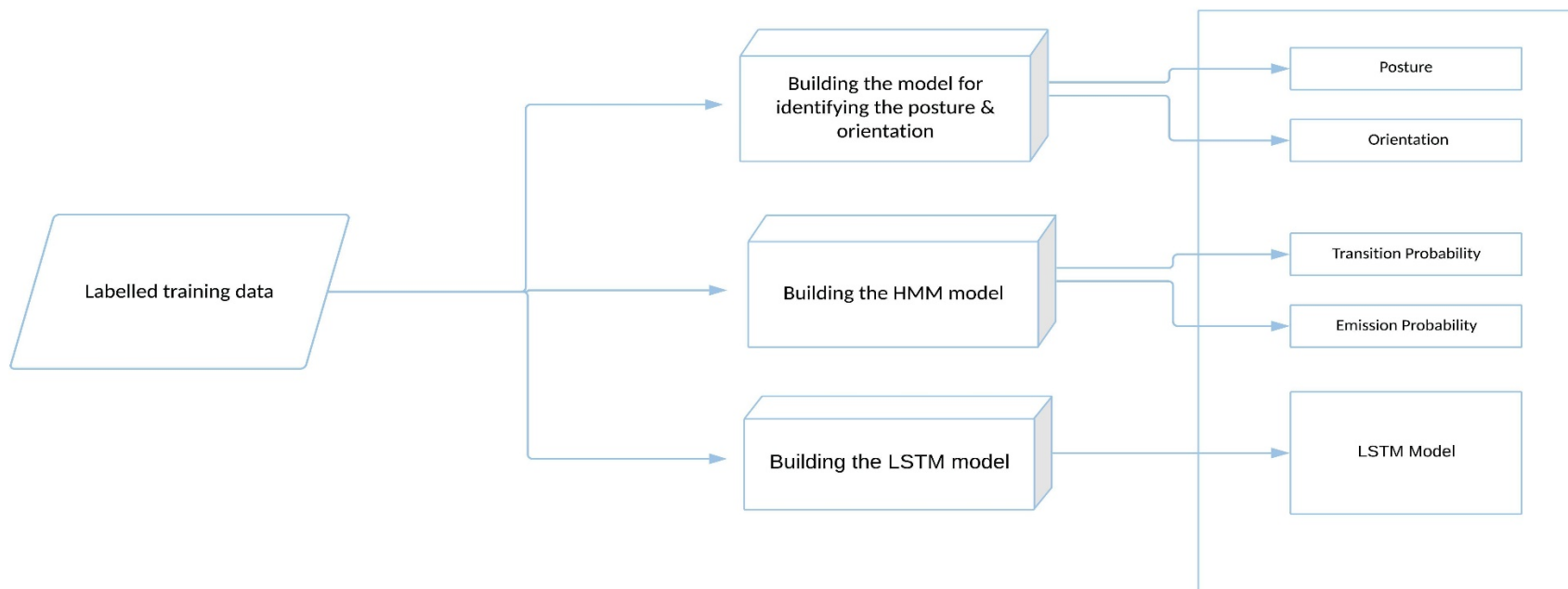


Figure 4. 19 HMM model for the four features. $A(t)$ denotes activity at time t and is the hidden state; $POL(t)$ represents the combination of posture, orientation, and location as the observed states.

Three methods are used to predict the activities, the HMM Forward algorithm, the HMM Forward-Backward algorithm, and the Long Short-Term Memory. The results for the methods are presented in Chapter 6.

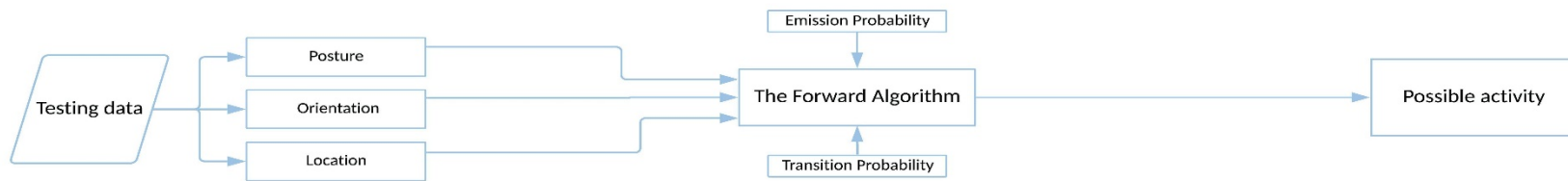


Step 1

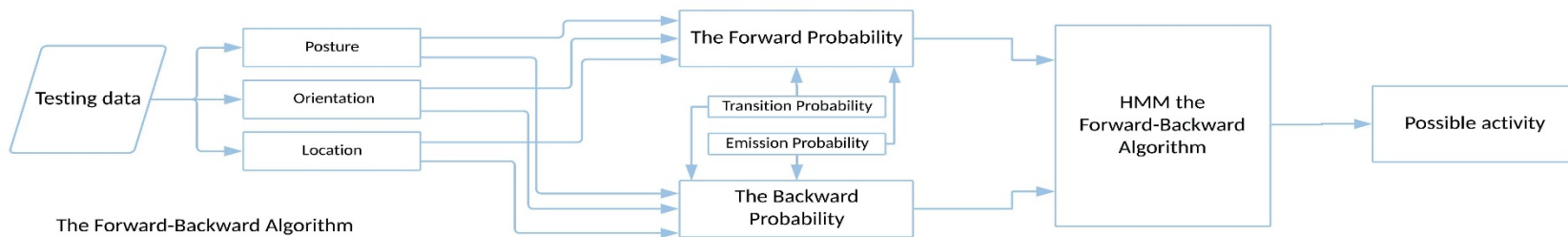


Step 2

Figure 4. 20 A general structure shows the first two major steps, gathering the data, creating and training the models



The Forward Algorithm



The Forward-Backward Algorithm



The LSTM

Step 3

Figure 4. 21: A general structure shows the last step, testing the proposed approaches

4.5.1 Summary of the Proposed Methods

The proposed approach uses HMM with different algorithms and classifiers. To implement the models, the data is labelled and then split into training and testing data. The labelling provides four high-level attributes, posture, orientation, location, and duration; in addition to the activity label.

To create the trained model, the recorded video clips are sampled to produce image sequence. The next step is to label the images based on the person's location, posture, orientation, time, and activity. The transition probability matrix and the emission probability matrix are calculated from the training data.

The next step is to feed the data into the CPM. The output from the CPM is normalised and converted into input feature vectors for posture and orientation classifiers. Two different methods for generating posture and orientation features are proposed. The location feature for the detected person is calculated from the CPM head coordinates.

The trained posture and orientation classifiers are used together with the calculated HMM emission and transition probabilities to predict activities (for HMM models). The HMM Forward algorithm, the HMM Forward-Backward algorithm, and the LSTM are used for that. The configuration for the LSTM is presented in Chapter 6.

Chapter 5 – Evaluation of High-Level Activity Identification

Chapter 5 presents the second objective of this work: to create a dataset that covers the range of activities that are focused on in this study and that have not been addressed in previous research, and then label the dataset according to the four high-level features used by the designed algorithms to identify these activities. This chapter also presents the experiments that were conducted to set up the network and record the video clips, and describes the tools that were used to carry out the experiments.

This chapter is structured as follows. Section 5.1 presents the reasons for creating a new dataset. Section 5.2 presents a complete description of the working areas and places where the video clips were recorded, as well as the details and specification of the cameras used. In the same section, the network and the selected servers in the working areas are also discussed, as well as the equipment security and privacy. Section 5.3 provides further details regarding the recorded dataset, the Portsmouth Activity Dataset, and the recorded activities and locations. Finally, Section 5.4 presents a comparison between the proposed dataset and some of the popular activities datasets.

5.1 Dataset Requirements

As mentioned in Chapter 3, the first objective of this study was to create a system that identifies the activities of daily living using high-level features including posture, orientation, location, and time (activity duration and recurrence). To evaluate the performance of this system, a dataset was required. As mentioned in Chapter 4, the selected methods use the aforementioned high-level features to identify the activities; thus, the dataset that was used was labelled based on these features.

The required dataset concerned the indoor instrumental activities of daily living that typically occur inside a house (Kempen & Suurmeijer, 1990; Waidmann & Freedman, 2006; Gobbens, 2018). These activities include cooking, using a toaster, using a microwave, using a sink, using a washing machine, using a fridge, preparing a meal, eating, working on a computer, looking for tools, looking for a file/document, preparing a hot drink, making a phone call, and walking. There are several such datasets available online for activity and action recognition; however, most could not be used for this work, for the following reasons.

The majority of available datasets were recorded with a single camera that was located either at a waist height location (body height) or head height location, and not in a top corner of a room, which would be the practical position for the camera (Palm, 1997). This affects the coverage area of the camera and makes it more susceptible to occlusion.

Another issue with the majority of the available datasets was that the recorded activities did not simulate an actual day, as they did not follow any scenario that is similar to daily human activities. Additionally, all the available datasets were labelled based on different types of features, and none used the high-level features needed for the methods used in the present study. Also, the available datasets recorded either staged activities or actions that are non-informative in daily activity recognition, such as standing and sitting.

In addition, some of the available datasets recorded all the activities in one location and not multiple realistic locations. For example, the MSR-Daily Activity 3D (Wang et al., 2012) recorded 16 activities in the same location (next to the sofa). This makes this popular dataset unsuitable for the methods used in this work, as the location is an essential feature.

In this study, the activity datasets are categorised into three main categories:

- Posture activities
- Low-level activities
- High-level activities

Here, posture activities (Laptev & Caputo, 2004; Blank et al., 2005; Rodriguez, Ahmed & Shah, 2008) are simple activities that can be performed using the upper and/or lower body within a period of time. For instance, walking, jogging, clapping, running, boxing, and hand waving. Some activities can be identified using posture movement, however this category has limitations in identifying location-based activities and high-level activities of daily living, which are more common for daily activities.

To overcome the limitations of the posture activities category, researchers created datasets based on low-level activities (Tenorth, Bandouch & Beetz, 2009; Rohrbach et al., 2012), such as reaching, picking something, lowering an object, opening and closing a door, and carrying something.

Low-level activities also depend on human body part movements within a window of time. However, they are sometimes linked to an object that is linked to a location. These activities are not useful in identifying the activities of daily living. For example, identifying an act of reaching-up, taking something from one place to another, using a mouse, or chopping, is not sufficient, alone, in determining what the detected person is doing. In addition, in some of these activities, there is no sense of location.

Other researchers have overcome these issues and worked on identifying high-level activities. High-level activities are a combination of low-level activities and posture activities. The MSR-Daily Activity 3D dataset (Wang et al., 2012) is a popular activity dataset that focuses on high-level indoor activities. Another group of researchers considered a different approach and combined posture activities, low-level activities, and high-level activities to produce the SPHERE-H130 action dataset (Tao et al., 2015).

However, neither of the aforementioned datasets could be used in the present work, as the activities recorded in the two datasets mentioned above are performed in a single location. Also, these high-level activity datasets focus on different activities, as presented earlier in Chapter 2, while the present study is targeting ADL and IADL, which are most likely to occur in an indoor environment (Kempen & Suurmeijer, 1990; Waidmann & Freedman, 2006; Gobbens, 2018).

Another concern about the available datasets was the camera viewing angle, due to the location of the cameras or the number of cameras used. In particular, the cameras were located either at a waist height location (body height) or head height location. Neither of these locations would be practical in a real-life home, as the cameras would either get in the way of the occupants or will often be occluded. Top corners of a room would be more practical positions. Some researchers have used multiple cameras in their work. For instance, in the TUM dataset (Tenorth, Bandouch & Beetz, 2009). In the TUM kitchen dataset, the researchers focused on low-level activities that took place in the kitchen. Another group of researchers overcame the limited viewing problem by using three cameras at head-height locations and developed the Multiview 3D Event dataset (Wei et al., 2013). However, this consisted of just low-level activities recorded in unrealistic locations (a lab).

A final limitation of the publicly available datasets is that all of them used different features to those required for this work. The methods selected for this study use high-level features to identify high-level activity, and there is no existing publicly available dataset labelled based on the features required for these methods.

The reasons for creating a new dataset can be summarised as follows:

- The requirement for a dataset with more realistic viewing angles so that researchers can investigate the use of the existing security monitoring cameras for assisted living.
- The requirement for using multiple cameras to record activities, which is useful to researchers who want to assess the effect of the camera viewing angle on identifying the activity, and who require a minimum number of cameras to identify all activities that occur in the location.
- The requirement to have different labelling, making the dataset useful to a broader range of researchers, such as those looking to identify posture, orientation, and activity.
- The requirement to have realistic locations for the performed activities, as not all the available datasets were recorded in realistic locations.
- The requirement to cover a different range of activities, as:
 - Some of the available datasets focused on posture activities, i.e., body movement over time.
 - Some of the available datasets recorded low-level activities.
 - The available datasets that recorded high-level activities did not consider the activities that the present work is seeking to identify.
 - The available datasets that recorded high-level activities were recorded in one location or a few locations. This means they are not useful for this study, as the selected methods consider the location as an essential feature.

The proposed new dataset, the Portsmouth Activity Dataset (PortAD), covers multiple realistic locations: a kitchen in an average house, and a working home office/lab. In these locations, multiple cameras were installed in the top corners of the room to record all the activities occurring in the location.

The PortAD activities are different from the activities in other datasets, as it was explicitly designed to mimic the normal life behaviour of a person inside a kitchen or a home office. The cameras were installed in the corners of the rooms at the ceiling level. This is an ideal location for installing cameras to cover the maximum area according to security experts (Palm, 1997). Multiple activities were recorded in various locations within each room, and these activities are selected based on the ADL and IADL that were most common in the house (Kempen & Suurmeijer, 1990; Waidmann & Freedman, 2006; Gobbens, 2018). PortAD did not record any bathing, grooming, dressing, or undressing activities due to privacy reasons; full details about the activities recorded will be provided later in the chapter.

5.2 Experimental Environment

PortAD covered two areas: the Port-Eco House (PEH) and a home office; more details about these areas are given in the following sections.

5.2.1 Port-Eco House

In this subsection, full details about PEH where the video clips were recorded are presented. The Port-Eco House is an experimental research facility provided by the University of Portsmouth for this work, shown in Figure 5. 1.



Figure 5. 1 Port-Eco House

The PEH is a regular two-storey house with three bedrooms, a living room, a kitchen, and a control room, where data processing servers are located. The house is equipped with smart sensors and appliances that are installed throughout the house. In each room, four Ethernet ports and four power outlets (sockets) are located in each corner at the ceiling level, and in one Ethernet port and a power socket is located in the middle of the, as shown in Figure 5.2.

These points are designed to allow cameras to be installed in the corresponding locations, which means five cameras can be easily installed in each room.



Figure 5. 2 Ethernet ports and power socket in a corner of a room.

The control room is on the ground floor and contains data processing computers and networking equipment, shown in Figure 5. 3. A Cisco network router connects the PEH private network to the University network. Local switches are used to connect all the PEH data ports for the cameras.

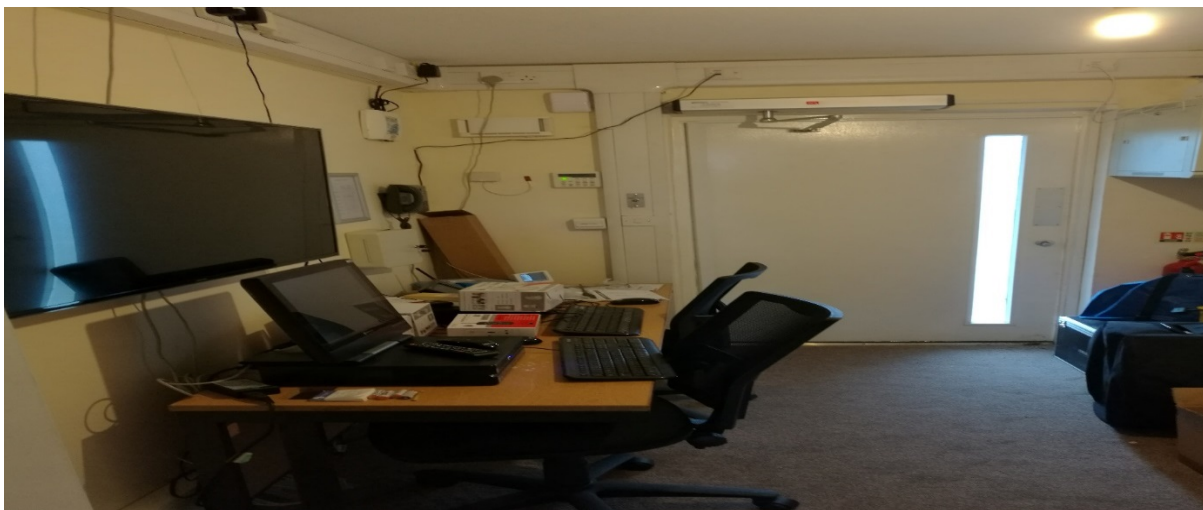


Figure 5. 3 The control room in PEH

In addition, the control room can be accessed from outside the house via a separate entrance without disturbing the residents of the house. All the servers and video recording are stored in this room, and only authorised personnel are permitted entry.

5.2.2 Home Office

The office is a computer lab. In the lab are multiple computers, a tool cabinet, a filing cabinet, and a table with a kettle to prepare hot drinks, shown in Figure 5. 4.



Figure 5. 4 Home office



The office has a secure access to ensure that only authorised individuals can enter. Cameras are installed in two corners at the ceiling level, covering the total office area. The cameras are connected via Ethernet to a local switch, and from there to the office DHCP server, which assigns IPs to the cameras and the computer. The recording process was performed on one of the university computers. To maintain a high level of privacy and data security, the computer was not connected to the campus network.

5.2.3 Cameras

Regular IP cameras were used to record the dataset video clips, rather than an expensive heat-vision cameras system, depth camera system, or fisheye camera system. The idea was to ensure the system would be available to everyone and could be installed in all premises, by using cost-effective solutions. Two models of Foscam IP cameras were used to create PortAD.

The two selected camera models were the Foscam FI9831W and the Foscam FI8909W. The FI9831W has slightly higher specifications than the FI8909W. Full details about the selected cameras are shown in Table 5. 1.

Table 5. 1 Specification for the two selected camera models (Foscam)

	FI8909W	FI9831W
		
Display resolution	300k Pixels	1.3 Megapixels (1280 x 960)
Frame rate	15 fps (VGA),30 fps (QVGA)	30fps
Lens type	f:2.8mm, F:2.4	f:2.8mm, F:2.4
Horizontal view angle	60°	80° (PT : H=300°, V=120°)
IR (night vision)	Yes	Yes
IR range	5m (16.4 feet)	8m (26.2 feet)
Ethernet	Yes	Yes
Wireless	Yes	Yes
Built-in microphone	Yes	Yes
PTZ camera	N/A	PT only

Initially, both camera models were used for the work, to check if changing the resolution and quality would affect the system. The proposed algorithms achieved the same results on both selected camera models. This meant it was not necessary to use the high-resolution camera for the work, thus lowering the cost and improving network traffic efficiency. However, to create this dataset, the high specification FI9831W model was used.

5.2.4 Network and Servers

Two private networks for the cameras were used in this work, to ensure the privacy and security of the recording process. One network was used for the wireless connections, and the second for the wired connections, as shown in Figure 5. 5. In addition, a private DHCP server located in the house assigned IPs to the cameras via Ethernet and Wi-Fi. Also, the system was configured to limit the number of Internet Protocol (IP) addresses, based on the Media Access Control (MAC) addresses for the cameras, to prevent new unauthorised devices from entering and connecting to the private network.

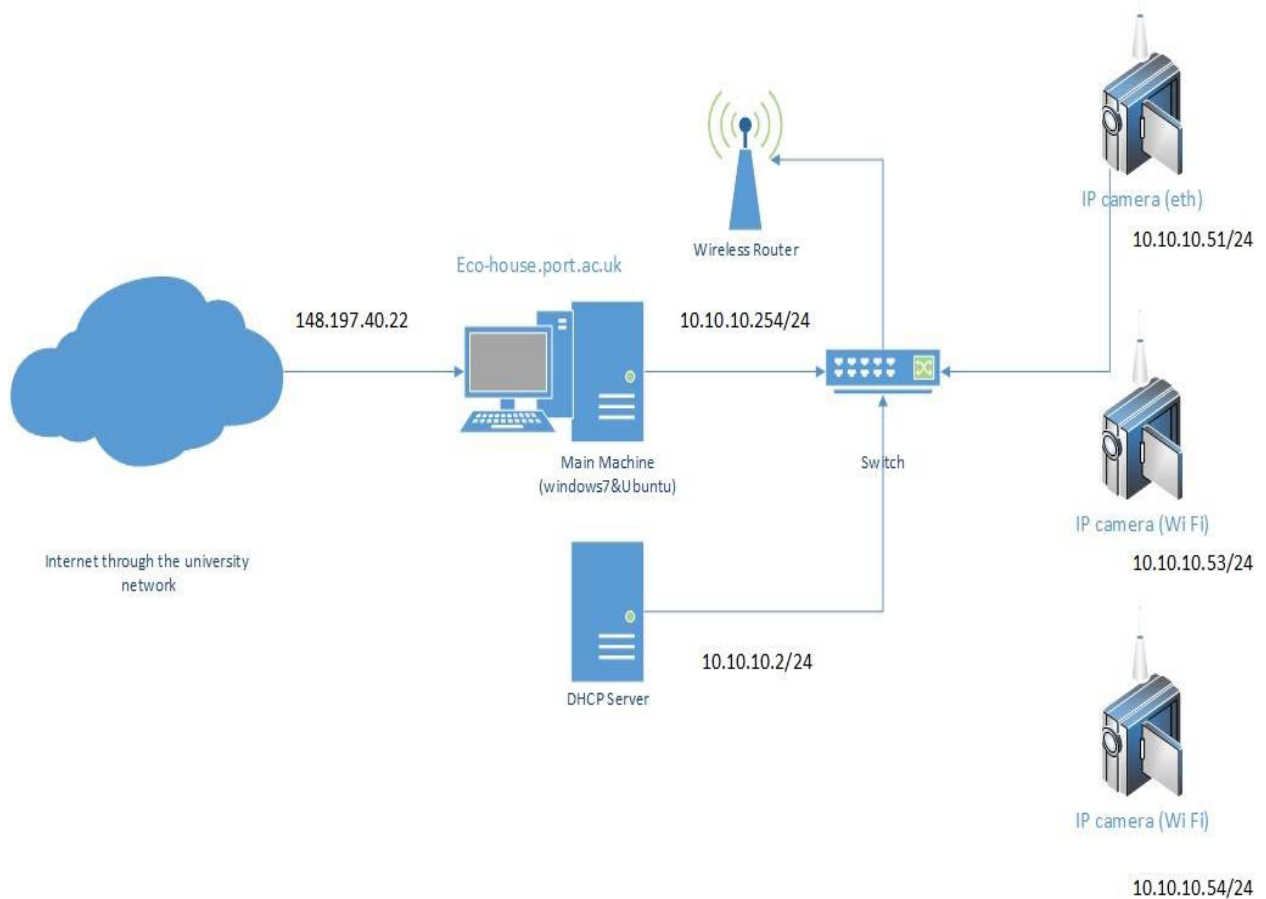


Figure 5. 5 The camera network for the PEH

Two servers, located in the PEH control room, were used for the experiments. The first was the DHCP server that assigned IPs to the cameras. The DHCP server was a dual boot machine with Intel i5 processor, 8GB of RAM, and 500GB of storage that ran Windows 7 and Ubuntu 16.04 LTS operating systems.

The second (main) server contained a large hard drive to store the videos from all the cameras. The main server was also a dual boot computer, with Intel i5 processor, 8GB of RAM, and 4TB of storage and again ran Windows 7 and Ubuntu 16.04 LTS. Due to the large partition size, the GUID Partition Table (GPT) was used instead of the Master Boot Record (MBR), as the MBR cannot identify hard drive partitions that are larger than 2 terabytes.

To access the house data from outside the PEH, and to prevent unauthorised access to the server, a Virtual Private Network (VPN) was used. OpenVPN was installed and configured in the server, where the videos are stored. The VPN connection provided a secure link to

the server in the house and allowed only those with the necessary authentication to access the network.

5.2.5 Privacy and Security

The system privacy and security are two different but related subjects, which sometimes overlap in this work. Therefore, these issues are discussed together in this subsection.

Security is a vital requirement in the suggested system design. The proposed network was designed to follow all standard network security protocols, and can be improved in the future if needed. The camera network was part of the private house network. The system was set up in a private house, and was installed on a personal computer with antivirus software and a firewall installed, activated, and enabled.

A Cisco router was also used to separate the house network from the rest of the University campus network. Only authorised IP and MAC addresses were permitted to connect to the network. Access control was limited to specific users with passwords and log files that recorded all the instances of access to the system. In addition to the passwords and routers, a Secure Sockets Layer (SSL) connection was required to access the system from a different network. A Virtual Private Network (VPN) with encryption was also used, with a log file showing full details regarding those who logged onto the system. OpenVPN was used with OpenSSL, a security library for encrypting data and control channels that uses 256-bit encryption.

To maintain the high level of privacy for the detected person, all the cameras were installed in the common house areas (not in the bathroom or the toilet). The system processes the data inside the monitored person's house, and the videos are saved on a computer that is installed at the same location. Also, only authorised personal with access permission can access the data. In addition, all the recording takes place in the main server, there is no recording in the cameras themselves. In the PEH, for the wireless connectivity, WPA2 PSK/AES was enabled in the cameras and the wireless router in the house.

Privilege user control was provided, and different levels of user groups were assigned for accessing the data, the computers, and the cameras. Some users were given admin rights that allow access to all the data and hardware; others had access to the camera live feeds only,

enabling them to see if something has happened to the detected person in case of an emergency, with no access to the recorded data file.

The need for a monitoring system was a significant issue. Therefore, the designed network followed the standard level of security and privacy, so the detected people could perform their daily activities safely. In the next ten years, people will become more familiar with cameras and monitoring systems, and will become more educated (familiar) about camera systems and privacy, as cameras become more ubiquitous. This will partially solve the camera privacy issue.

5.3 Experimental Scenarios

In PortAD, one subject performed all the activities in all the locations. To add variation to the recording, the subject was wearing at least one different piece of clothing each day. The subject (the PhD student) remained in the house for the allowed working hours and acted as if he was living in the house. The subject did not actually live in the house due to University regulations and legal reasons. In the recording scenarios, the subject was vigilant in taking care not to perform the activities in a way that would improve the performance of the proposed methods.

The videos were recorded in MP4 format with a resolution of 1280 x 720, a data rate of 1805 kbps, and a frame rate of 30 frames/sec.

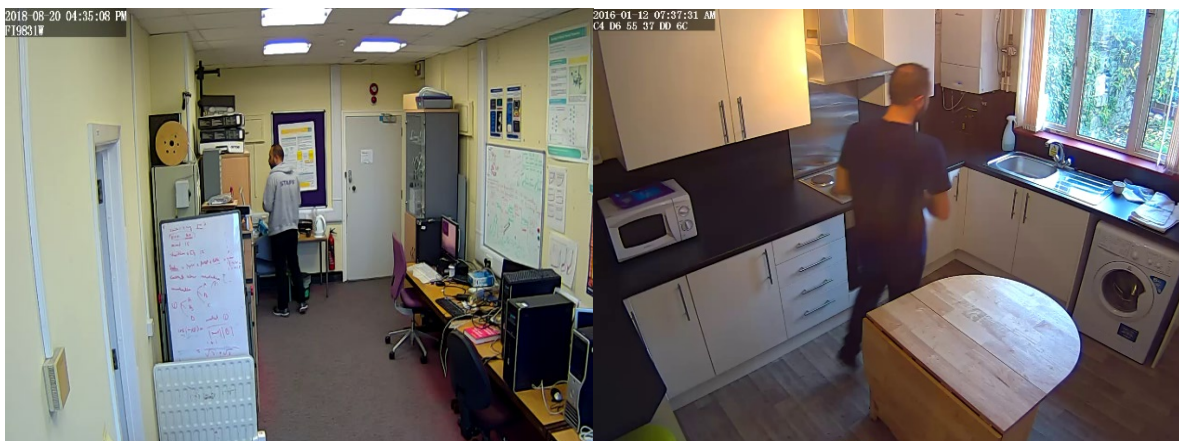


Figure 5. 6 Samples of the data in the kitchen and home office

In PortAD, two main categories of human activities of daily living were covered: the basic activities of daily living (ADL), and the instrumental activities of daily living (IADL). The basic

ADL, or self-care activities, are those conducted by the individual themselves (Huang et al., 2015; Mlinac & Feng, 2016). Below are the basic ADL included in PortAD:

- Ambulating
 - Standing
 - Sitting
 - Bending/bowing
 - Walking
- Eating

Some basic activities were used as features in PortAD, such as standing, sitting, and bending/bowing, and labelled accordingly, as the proposed methods required these high-level features as an input to predict the high-level activities.

Instrumental activities are defined by their ability to enable the person to live independently and to perform daily activities. In PortAD, 12 instrumental activities were recorded:

- Working on a computer
- Making a phone call
- Using a tool cabinet
- Using a filing cabinet
- Using a kettle
- Cooking
- Using a washing machine
- Using a toaster
- Using a microwave oven
- Using a refrigerator
- Using a sink
- Preparing a meal

5.3.1 Home Office Activities

This section covers the activities that occurred in the home office. The home office activities copied the human activities that typically take place in an office, for five days from Monday to Friday (i.e., weekdays), a full working week, as shown in Table 5. 2. These activities were selected to cover the ADL and IADL that are commonly performed in an office.

The duration of PortAD activities depended on the activity itself. For short activities, such as using the kettle, the tool cabinet, or the filing cabinet, walking, and being on the phone, these activities were performed based on the actual activity length. These activities were executed and recorded based on their similar execution (duration) times, as the activities were performed with all details (i.e., in full length) several times each day. The timing of these activities varied from less than a minute to six minutes, depending on the activity. The activities were performed and recorded without shortening them, to imitate an individual's daily activity in an office.

For the 'working on a computer' activity, the full activity duration was not always recorded, as the length of this activity typically ranged from 5 to 80 minutes. The posture, orientation and location for working on a computer were sitting on a chair, facing the computer, and a location near the computer, as shown in Figure 5. 7. The key difference for this activity is the activity duration, which is the reason for shortening the recording for this activity. Therefore, the activity 'working on the computer' must be lengthened based on the timing in Table 5. 2, or any other suitable timing to model the real-life durations. This can be done by evenly replicating frames for the activity.



Figure 5. 7 The detected person working on the computer

PortAD covered the possible sequences of activities for five days, Monday to Friday. Different orders and durations of activities were recorded. For example, starting the day by working on the computer or making a hot drink, as can be seen in Table 5. 2. The timing of the recorded

activities was based on office working hours. For example, Friday was shorter than other days; the full working week started on Monday at 09:00 and ended at 17:00 each day, apart from on the Friday when it ended at 16:00.

Two cameras located in the corners at the ceiling level were used to record the activities in the office. These two cameras covered all the locations and objects in the room. Six different activities were performed in different orders, multiple times each day. These activities were: working on the computer, searching for a tool, searching for a file/documents, making a hot drink, using the phone, and walking.

In the office area, in the middle of the day, there were times where no person was in the office, for instance to take a lunch break or a meeting. At these times, the subject left and returned to the office from the same door. For both cameras, the duration of the time when the office was empty was less than 5% of the recording time.

In the recorded video clips, the three different postures (standing, sitting, and bowing/bending), and four different orientations (facing, away, left, and right) were covered, in addition to multiple locations and activities.

The sequences for these activities were based on observing the behaviour of people in the office that used to record these activities and based on the working hours specified in the office policy of the University of Portsmouth.

Table 5. 2 Home office activities, with the day of the week, time of day, activity name and duration

Monday		Tuesday		Wednesday		Thursday		Friday	
09:00	Computer	09:00	Computer	09:00	Computer	09:00	Kettle	09:00	Computer
09:20	Kettle	09:20	Kettle	09:20	Kettle	09:05	Computer	09:30	Computer
09:25	Computer	09:28	Computer	09:30	Computer	09:30	Computer	09:40	Kettle
09:45	Phone	09:45	Tool cabinet	09:45	Phone	10:00	Computer	9:46	Computer
10:00	Computer	10:00	Computer	10:00	Computer	10:10	Phone	10:10	Phone
10:30	Computer	10:10	Kettle	10:20	Kettle	10:30	Computer	10:30	Computer
11:00	Computer	10:30	Computer	10:30	Computer	11:00	Computer	11:00	Computer
11:30	Computer	11:00	Computer	11:00	Phone	11:25	Phone	11:25	Kettle
11:50	Filing cabinet	11:30	Computer	11:30	Computer	11:30	Computer	11:30	Computer
12:00	Computer	11:50	Filing cabinet	11:50	Tool cabinet	11:50	Filing cabinet	11:50	Filing cabinet
12:30	Break	12:00	Computer	12:00	Computer	12:00	Break	12:00	Break
13:00	Break	12:30	Computer	12:30	Phone	12:30	Break	12:30	Break
13:30	Computer	13:00	Break	12:35	Computer	13:00	Computer	13:00	Computer
14:00	Computer	13:30	Break	13:00	Break	13:25	Kettle	13:25	Kettle
14:15	Phone	14:00	Computer	13:30	Break	13:30	Computer	13:30	Computer
14:20	Kettle	14:15	Phone	14:00	Kettle	14:00	Computer	14:00	Computer
14:30	Computer	14:20	Filing cabinet	14:10	Computer	14:15	Phone	14:10	Tool cabinet
15:00	Computer	14:30	Computer	14:20	Filing cabinet	14:20	Tool cabinet	14:30	Computer
15:10	Tool cabinet	14:35	Kettle	14:30	Computer	14:30	Computer	15:00	Computer
15:30	Computer	15:00	Computer	14:35	Kettle	15:00	Computer	15:10	Phone
15:50	Tool cabinet	15:10	Tool cabinet	15:00	Computer	15:10	Tool cabinet	15:30	Computer
16:00	Computer	15:30	Computer	15:10	Filing cabinet	15:30	Computer	15:50	Filing cabinet
16:10	Kettle	15:50	Tool cabinet	15:30	Computer	15:50	Filing cabinet	16:00	Computer
16:30	Computer	16:00	Computer	15:50	Filing cabinet	16:00	Computer		
17:00	Computer	16:30	Computer	16:00	Computer	16:10	Kettle		
		17:00	Computer	16:15	Filing cabinet	16:30	Computer		
				16:30	Computer	16:35	Phone		
				16:40	Filing cabinet	17:00	Computer		
				17:00	Computer				

5.3.2 Common Domestic Activities

The second working area was the PEH kitchen. Three days of activities were recorded in the kitchen, for the three main meals: breakfast, lunch, and dinner. This is shown in Table 5. 3.

Nine activities were performed in the kitchen, covering ADL and IADL: cooking, eating, using the sink, using the washing machine, preparing a meal, using the microwave oven, using the fridge (refrigerator), using the toaster, and walking. Similarly to the home office, different combinations and sequences of activities were covered to provide more variety to the dataset.

In this dataset, some activities were performed in full-length several times (i.e. using the actual duration to perform the activity). These activities included using the sink, using the washing machine, using the microwave oven, using the refrigerator, and walking. In regard to cooking, eating, and preparing meals. These activities are subjective as they depend on the person, the type of meal, eating habits, and speed. These three activities can be easily lengthened when needed, by increasing the number of frames (using replication) for the activities that need to be lengthened.

In the kitchen, four cameras were used. These cameras were located in the corners of the kitchen area at the ceiling level, to cover all locations in the room. The same person performed all the activities in the PEH kitchen for three days, wearing different clothing each day. The PEH kitchen was a small working area, and there were some occluded zones because of the viewing angles and locations of the used cameras.

Nine activities were performed in the PEH kitchen. These activities have three postures (standing, sitting, and bowing/bending) and four orientations (facing, away, left, and right). Three meals were prepared in front of the cameras, and some activities related to the kitchen location, such as using the washing machine, were performed and recorded (Arnold, Ball, Duncan & Mann, 1993). These activities were selected based on people's average daily activities.

Table 5. 3 Scenarios for three days in the PEH kitchen, for three meals

	Day one	Day two	Day three
Breakfast	Toaster	Sink	Microwave
	Microwave oven	Microwave	Kettle
	Sink	Kettle	Toaster
	Microwave oven	Preparing meal	Sink
	Toaster	Sink	Microwave
	Cooking	Eating	Toaster
	Preparing meal	Preparing meal	Preparing meal
	Dining table	Microwave	Eating
	Preparing meal	Sink	Preparing meal
	Sink	Washing machine	Sink
Lunch	Sink	Kettle	Microwave
	Fridge	Microwave	Sink
	Microwave	Dining table	Cooker
	Cooker	Kettle	Toaster
	Toaster	Microwave	Washing machine
	Preparing meal	Fridge	Sink
	Dining table	Preparing meal	Microwave
	Preparing meal	Dining table	Toaster
	Toaster	Preparing meal	Preparing meal
	Cooker	Toaster	Dining table
	Microwave	Sink	Preparing meal
	Sink	Fridge	Sink
			Fridge
Dinner	Toaster	Cooker	Sink
	Cooker	Microwave oven	Cooker
	Washing machine	Cooker	Dining table
	Microwave oven	Sink	Sink
	Cooker	Cooker	Toaster
	Sink	Sink	Cooker
	Cooker	Preparing meal	Microwave
	Dining table	Dining table	Cooker
	Cooker	Preparing meal	Toaster
	Preparing meal	Cooker	Microwave
	Microwave oven	Microwave oven	Preparing meal
	Preparing meal	Dining table	Dining table
	Dining table	Sink	Preparing meal
	Sink	Fridge	Washing machine
Washing machine			

5.4 Comparison with Existing Datasets

PortAD recorded all ADL and IADL that the present study is targeting, using practical camera locations that are most likely to be used if the cameras are installed in a house or an office. According to security experts, the corner location at the ceiling level is the best location to install cameras (Fennelly, 2016; Norris & Moran, 2016), as it provides a wider viewing angle compared to other locations, enabling the full area to be covered with a minimum number of cameras. This will help researchers who are investigating the use of security camera systems for assisted living, and vice versa.

In PortAD, all activities were recorded in realistic locations. For instance, cooking and using the microwave oven took place in the kitchen and not in a lab. Multiple cameras were used in each area to cover all locations. Adoption of this method will help researchers seeking to evaluate the effectiveness of different camera viewing angles in identifying activities and determining the minimum number of cameras required to correctly identify activities.

Multiple scenarios were covered for each meal to provide different sequences of activities, as shown in Table 5. 3. An example of the angles covered by the PortAD dataset cameras is shown in Figure 5. 8.

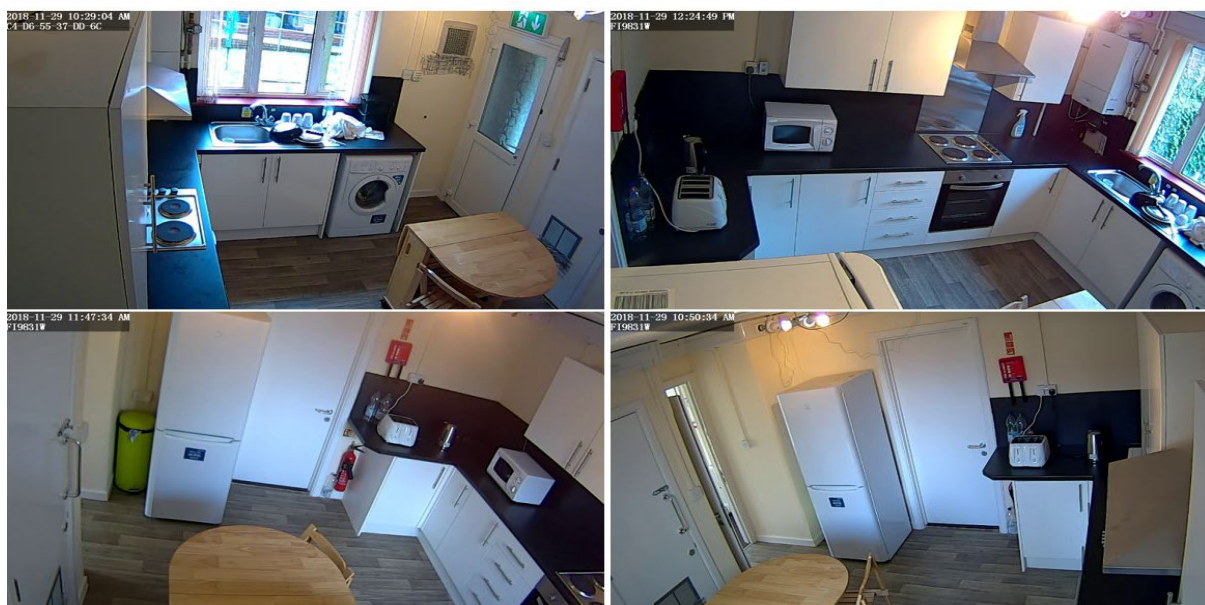


Figure 5. 8 Camera views for the PortAD dataset

A comparison between the PortAD dataset and other publicly available datasets is presented in Table 5. 4 below.

Table 5. 4 Summary of popular activity recognition datasets and PortAD

Datasets	Classes (activities)	No. of subjects	No. of cameras	Location	Type of activity	Sensor type	Camera location	Year
KTH	6	25	1	single/ outdoor	low-level	grey scale	varies	2004
Weizmann	10	9	1	single/ outdoor	low-level	RGB	outdoor	2005
TUM Kitchen	10	4	4	multiple	low-level	RGB	top corners	2009
MPII Kitchen	65	12	1	multiple	low-level	RGB	top centre	2012
SPHERE-H130	13	5	1	single	high-level	RGB-D	top centre	2015
UR Fall+Action	2+4	5	2	multiple	sudden/ low-level	RGB-D	body height	2014
MSR-Daily Activity 3D	16	10	1	single	high-level	RGB-D	body height	2012
Multiview 3D	8	8	3	small number of locations	low-level	RGB-D	head height	2013
CAD-60	12	4	1	small number of locations	low-level	RGB-D	body height	2012
CAD-120	10	4	1	small number of locations	high-level	RGB-D	body height	2013
PortAD	14	1	2 - 4	multiple locations	high-level	RGB	top corners	2019

The distinguishing characteristics of the PortAD dataset can be summarised as follows:

- 1- It looks at different features. The PortAD data is labelled based on four high-level features: posture, orientation, location, time, and activity.
- 2- It uses realistic locations. All the activities were recorded in realistic locations in a way that reflected people's daily activities.
- 3- It records 14 activities, four different orientations, and three postures.
- 4- It shares some activities with other datasets, such as walking and preparing a meal, but the specific combination of 14 activities has never been used before.
- 5- It uses practical camera locations. Unlike other high-level activity datasets, PortAD cameras are installed in the corners at the ceiling level.

- 6- It uses several cameras per location. Compared to other high-level activities datasets, the PortAD dataset has wider coverage areas because multiple cameras are used.
- 7- In the PortAD dataset, one subject performed the activities in all locations. This can be increased in the future.
- 8- All the activities in the PortAD dataset were performed during the day. Night-time activities can be added in the future.

Chapter 6 Effectiveness of Features and Fusion Methods

This chapter describes the experiments that have been carried out to evaluate the proposed algorithms and the rationale behind them in Section 6.1. The datasets that are used to evaluate the work and the configurations for the selected methods are presented in Sections 6.2 and 6.3, respectively. The performance metrics used for evaluation are discussed in Section 6.4, and the hardware and software that were used to implement the work are described in Section 6.5.

The results for each proposed approach are presented in Section 6.6, addressing the single feature approach, temporal and spatial features, three and four features. Full analysis of the results is presented in Section 6.7. Finally, Section 6.8 provides a comparison of the effectiveness of all the selected features and methods.

In this work, two datasets are used to evaluate the work. PortAD is used to evaluate the identification of activities of daily living. The UR fall dataset (Kwolek & Kepski, 2014) is used to evaluate the selected features and methods for identifying fall.

6.1 Conducted Experiments

The experiments have been designed to evaluate the approaches presented in Chapter 4. For identification of ADL, the main focus is on assessing the effectiveness of the four proposed high-level features (spatial, temporal, posture, and orientation). The performance of ADL identification depends on the accuracy of detecting the high-level features from video data in the first place. Therefore, looking at the end results only may be misleading, as poor ADL identification performance may be a consequence of poor feature detection, rather than the low effectiveness of the feature itself. To separate these two aspects, the experiments to assess the features effectiveness are carried out using manually labelled features (i.e. true feature values) as well as using detected features (i.e. using posture and orientation detection methods described in Chapter 4). Since a number of different methods are proposed in Chapter 4 for detecting posture and orientation, the performance of these methods needs to be evaluated first, to select the best method to provide features for subsequent activity identification.

Based on these considerations, the following groups of experiments have been conducted:

1. Evaluation of the effectiveness of the selected high-level features for identification of ADLs using manually labelled (true) features:
 - a. Using location only
 - b. Using location and duration
 - c. Using location, duration, and posture
 - d. Using location, duration, and orientation
 - e. Using all four features – location, duration, orientation, and posture
2. Evaluation of the performance of different approaches to detecting posture and orientation features.
3. Evaluation of the effectiveness of using location, duration, orientation, and posture for ADL identification based on the features detected from video data.
4. Evaluation of the validity of the assumptions underlying the proposed conditional probability-based approach for combining spatial, temporal, posture, and orientation features using HMMs.

For each of the experiments in groups 1 and 3 above, the corresponding algorithms described in Chapter 4, including threshold-based, HMM-based, and LSTM, are applied and evaluated. Experiments in group 2 consist of evaluating different classifiers and different ways for representing body part coordinates (normalised coordinates and pair-wise distances between body parts). Finally, experiments in group 4 are used to look into implications of using the conditional probability-based method for calculating HMM observation probabilities, when combining all four high-level features (see Section 4.4).

6.2 Data Preparation

The experiments are carried out using two labelled datasets. The first one is the PortAD dataset described in Chapter 5. The second dataset is the UR fall dataset (Kwolek & Kepski, 2014). In the PortAD dataset, two locations were used: the kitchen and the home office. The home office data contain videos from two cameras located in different corners of the room. The kitchen data contain videos from four cameras located in each corner of the room to improve the area coverage. The full details are provided in Chapter 5.

The home office data were split into two parts: approximately 70% of the selected data was allocated for training; the remaining 30% was used for testing. As explained in Chapter 5, the videos cover five simulated weekdays, which share the same activities, but differ in activities

sequences and durations. Due to the workload associated with labelling the data, only three out of five available days have been manually labelled with feature and activity labels. Therefore, two of these days have used as the training data and one day as the testing data to achieve the desired 70/30 training/testing split.

The same approach for selecting the training and testing data was used for the kitchen videos, resulting in a similar 70/30 split between training and testing data. Two fully labelled days were selected for training and one for testing.

In the UR fall dataset (Kwolek & Kepski, 2014), 30 short video clips were recorded for falls from the standing and sitting positions. These falls were performed by 5 subjects. Two Kinect sensors recorded these falls one was located at the top centre location, and the other one was located at a side (body height) location. The system used to record the data combined the readings from accelerometers and depth images sensors and achieved 98.33% accuracy, 96.77% precision, 100% sensitivity, and 96.67% specificity. Although two Kinect sensors were used to create the dataset, only one Kinect sensor is used in this work. The RGB video data from the side Kinect sensor is used. This sensor location is more suitable for the selected CNN compared to the one at the top centre location, as the used CNN cannot extract the pose features from the sensor that is positioned at the top centre location.

The clips from the side Kinect sensor need to be converted to make it suitable for the proposed methods. Therefore, the frames from the 30 video clips in the dataset were labelled based on the high-level features (posture, orientation, location, and time), and activity. Four different postures have been differentiated and labelled in the data: standing, bending, sitting, and lying. Four orientation left, right, away, and facing. The labelled activities consist of sitting on a chair, walking, and falling. These activities took place at four locations, which are also labelled in the data. Twenty videos were used to train the model and ten videos to test.

In the PortAD dataset, individual frames have been similarly labelled with the three high-level features for the person in the frame (location, posture, and orientation) and the activity that was happening in that frame.

For HMMs, training data is used to estimate transition and emission probabilities. This can be done directly from a labelled frame sequence (using equations in Chapter 4). Thus, no further

pre-processing of the labelled data is required. For LSTM, each training instance consists of features for a frame sequence (rather than a single frame) as well the activity label for the last frame in the sequence. The length of the frame sequence (window size) in each training instance is the same and depends on the LSTM input layer shape. Thus, to prepare the training set for an LSTM, the full labelled frame sequence needs to be cut into training instances according to the window size. This is illustrated in Figure 6. 1. The input shape (windows size) of five frames was used for the presented results. This input shape was selected empirically as delivering the best performance on both the PortAD dataset and the UR fall dataset.

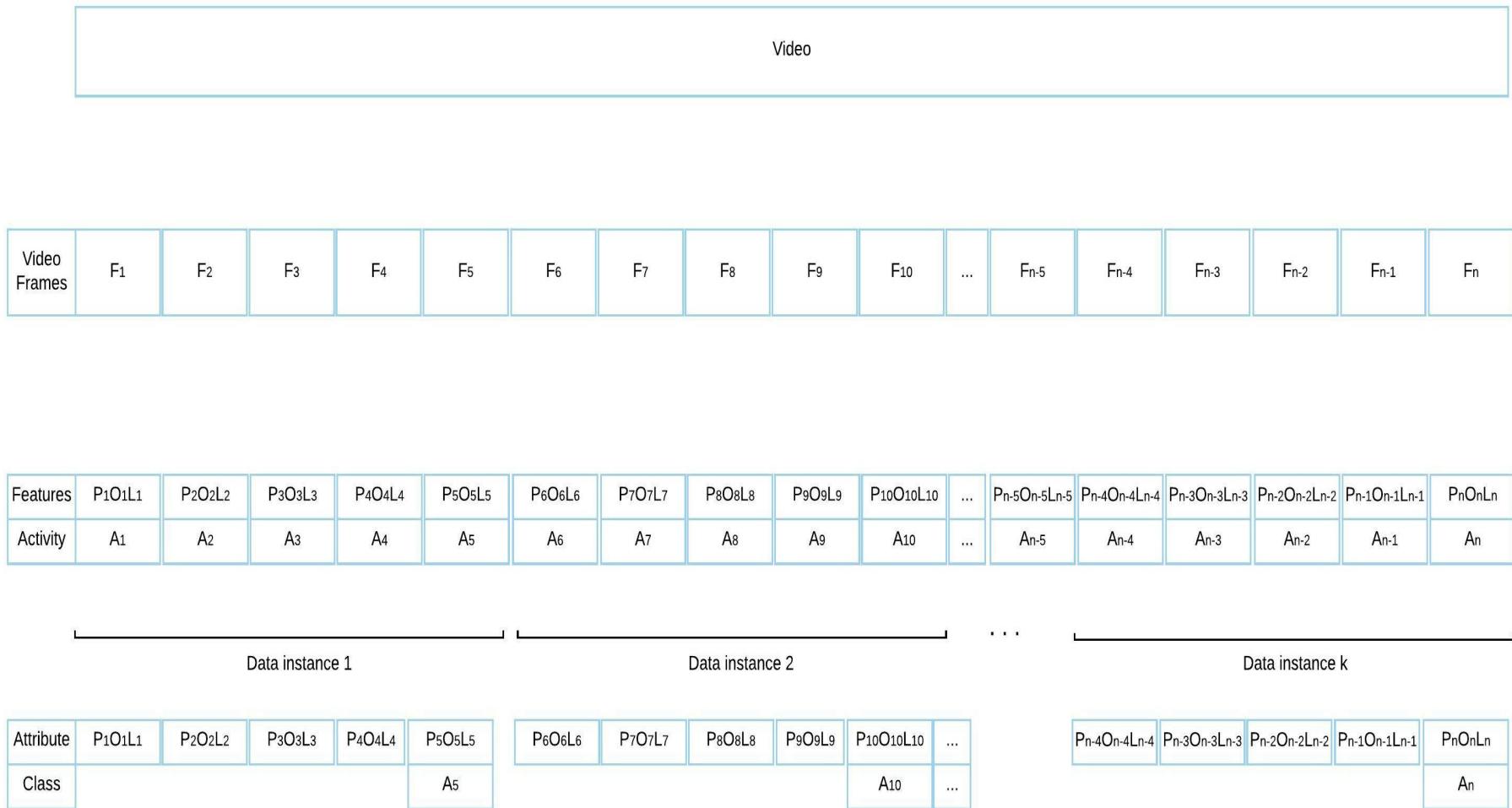


Figure 6. 1: LSTM window size for training and testing, when the window is equal to five (P=posture, O=orientation, L=location)

6.3 Algorithm Parameters Selection

Most algorithms presented in Chapter 4 include a number of parameters, which may affect the performance of each algorithm. Selecting values of these parameters were done either empirically by choosing the best performing value from a test range or based on the best practice used in the state-of-the-art research for that algorithm.

The threshold values for the fixed time threshold algorithm (Section 4.2.2) were tested in the range from 1 to 105 frames. It was observed that when the value exceeded 5 frames, the performance decreased. Therefore, the fixed threshold was set at 5 frames.

The activity-specific threshold for each activity is calculated as the minimum, maximum, and average duration for the given activity in the training data. These values can be calculated from the training data. The minimum threshold values ranged between 1 and 175 frames. The maximum adaptive threshold varied between 32 and 404 frames depending on the activity. The average adaptive thresholds ranged from 3 to 300 frames.

The LSTM training can use a number of different loss functions for the network weights adjustment. In this work, sparse categorical cross-entropy was used as the loss function as it achieved the best results when tested, compared to the mean absolute error, mean squared error, and the mean absolute percentage error. Also, it is suitable for the available labelled data.

An optimisation method needs to be selected for LSTM to assign weights based on training performance, as mentioned in Chapter 4. Adagrad (Duchi, Hazan & Singer, 2011), RMSprop (Hinton, Srivastava & Swersky, 2012; Ruder, 2016) and ADAM (Kingma & Ba, 2014) achieved the highest results on both PortAD and UR fall data compared to other optimisers. ADAM was selected out of these three optimisers as the LSTM optimisation method, because it achieved the highest performance in fewer training iterations. The values used for the ADAM optimiser were selected based on Kingma and Ba (2014) prior work.

The following configuration for the LSTM model achieved the best results for the datasets used in this work:

- Input layer shape (the number of inputs) is equal to the windows size (5 frames) multiplied by the number of high-level features used as follows:
 - When using spatio-temporal features, the number of features is 1 (location).

- When using spatio-temporal and a single pose-based feature, the number of features is 2 (location and either posture or orientation).
- When using spatio-temporal and both pose-based features, the number of features is 3 (location, posture, orientation).
- Hidden layers:
 - Hidden layer 1 consist of 64 neurons with the “*tanh*” activation function and 0.2 dropout function.
 - Hidden layer 2, consist of 64 neurons with the “*tanh*” activation function and 0.2 dropout function.
 - Hidden layer 3 is a 32 neurons dense layer with a 0.2 dropout function.
- Output layer is a “*softmax*” layer. The number of neurons in this layer is equal to the number of activities.

The number of training epochs for LSTM was set to 100.

When detecting a person’s location, the BS-KNN was used if the CPM did not detect the person’s head, as explained in Chapter 4. To avoid detection of unnecessary movements (small objects, changes in lighting, etc), a threshold was set for the object contour size to be larger than 4000 pixels for the VGA (640x480) resolution and larger than 6000 pixels for the 720p resolution videos. These threshold values have been selected after trying several different contour sizes on the used datasets.

A number of classifiers have been compared for posture and orientation detection (see Chapter 4). The configuration parameters for most of them have been selected based on the default values provided in the Weka implementation of these classifiers (Witten & Frank, 2002). For the Feedforward Neural Network, different numbers of hidden layers with different numbers of neurons have been tested. The number of hidden layers varied between 3 and 12. The number of neurons for each hidden layer ranged from 100 to 3000, with learning rate values from 0.1 to 0.001, and momentum between 0.2 to 0.9. The model that achieved the best performance on the PortAD and UR fall datasets was be used. It consists of three hidden layers, and the number of neurons in the hidden layers are 300, 250, and 150, respectively. The learning rate of 0.01 and momentum of 0.8 were used.

6.4 Performance Metrics

The algorithms' performance is assessed using accuracy, precision, recall, and F1 score. Precision, recall, and F1 score are typically used with binary classification tasks. Therefore, each metric was first calculated for each activity individually, as for a binary identification task. Then, an average across all activities was used to calculate the values for the precision, recall, and F1 score to obtain a single value for each metric.

For the fall detection performance, specificity is also used (i.e., the true negative rate). It evaluates the probability of non-fall, given that non-fall happened. Therefore, it will show how good the methods are at avoiding false alarms.

The performance of each method was measured separately on the data from each camera. In the results in Section 6.6, three values are presented for each metric: the minimum performance across all cameras; the maximum performance across all cameras; and the average performance across all cameras. Since each camera provides a different view of an activity scene, the minimum, maximum, and average values represent the worst, the best, and the average case performance, depending on obstructions and other limitations in the field of view.

6.5 Hardware and Software

The CPM was implemented and configured using Python with required libraries and tools (Duckworth et al., 2017), including OpenCV, NumPy, SciPy, math, time, utils, copy, glob, pandas, CSV, scikit-learn and Caffe (Jia et al., 2014). The specification for the machine that is used to run the CPM is Intel Core i7 6700, 32 GB of RAM, and Nvidia 1070 GPU. The classifiers were implemented using Weka (Witten & Frank, 2002), Knime (Berthold et al., 2009), and Python.

TensorFlow (Abadi et al., 2016), Keras (Chollet, 2015) and Python were used to implement the LSTM models. The code was run on the Google Colaboratory (Colab) (Carneiro et al., 2018; Bisong, 2019). Intel Xeon CPU, 25 GB of RAM, and Nvidia Tesla K80 from Google Colab were used to speed up the training process. The HMM code was written using Python and Matlab and implemented on the same machine as for the CPM.

6.6 Results

Figure 6. 2 and Table 6. 1 show the activity identification performance using the proposed methods on the PortAD dataset. Figure 6. 3 and Table 6. 2 present the fall detection performance for the proposed methods on the UR fall dataset. Figure 6. 4 and Table 6. 3 show the accuracy for detecting the posture and orientation using different classifiers and feature representation methods on the PortAD data. Table 6. 4 and Figure 6. 5 present the accuracy for identifying the posture and orientation on the UR fall dataset. Finally, Table 6. 5 shows Pearson correlation results for the three conditional probability assumptions used for estimating HMM emission probabilities when using all four high-level features (location, duration, posture, and orientation).

Table 6. 1: The activity identification performance on the PortAD dataset ('Min', 'Max', and 'Avg' correspond to the minimum, maximum, and average performance across different cameras; Fixed 5 f/sec = The fixed time threshold, when the threshold is 5 frames/sec; A-Min= Adaptive minimum threshold; A-Max= Adaptive maximum threshold; A-Avg= Adaptive average threshold; HMM-F= HMM with the Forward algorithm; HMM-FB =HMM with the Forward-Backward algorithm) Represent the highest values for each feature combination. Represent the highest values across all feature combinations

		Accuracy			Precision			Recall			F1 score		
Features	Method	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
Spatial	Location	76.13	89.01	96.18	67.00	85.83	96.00	70.00	85.43	96.00	68.00	85.80	96.00
Spatial and temporal. Spatial is determined from the true labels	Fixed 5 f/sec	77.57	90.52	96.16	70.00	88.17	96.00	70.00	86.50	96.00	70.00	87.17	96.00
	A-Min	65.16	76.00	85.43	65.00	79.50	90.00	65.00	76.83	86.00	65.00	78.11	86.90
	A-Avg	31.92	42.29	55.51	37.00	44.67	50.00	35.00	44.83	53.00	35.97	44.73	51.46
	A-Max	31.92	44.89	55.84	37.00	43.50	50.00	38.00	43.50	50.00	37.49	43.50	50.00
	HMM-F	81.63	91.42	97.18	84.00	90.83	97.00	70.00	83.33	93.00	72.00	84.17	94.00
	HMM-FB	82.69	91.45	97.40	82.00	90.17	97.00	71.00	84.00	94.00	74.00	84.67	95.00
	LSTM	77.76	90.68	96.50	72.00	87.00	96.00	72.00	84.83	92.00	69.00	84.00	92.00
Spatial, temporal and orientation. Orientation and spatial determined from the true labels	HMM-F	82.69	91.45	97.40	82.00	90.17	97.00	71.00	84.00	94.00	87.00	90.83	96.00
	HMM-FB	89.69	94.29	97.70	88.00	92.33	97.00	87.00	91.00	95.00	74.00	84.67	95.00
	LSTM	84.29	92.89	97.48	77.00	87.67	96.00	80.00	88.00	94.00	78.00	86.83	95.00
Spatial, temporal, and posture. Posture and location determined from the true labels	HMM-F	91.43	95.53	98.76	91.00	94.67	98.00	86.00	91.00	97.00	88.00	92.17	97.00
	HMM-FB	91.45	95.53	99.00	91.00	94.50	97.00	86.00	91.17	98.00	88.00	92.17	97.00
	LSTM	90.82	95.82	98.70	93.00	96.00	99.00	88.00	92.67	97.00	88.00	93.33	98.00
Spatial, temporal, posture and orientation. Posture, orientation, and spatial determined from the true labels	HMM-F	93.82	97.74	99.55	91.00	97.00	100.00	88.00	95.83	99.00	89.00	96.17	99.00
	HMM-FB	93.82	97.82	99.78	92.00	96.83	100.00	88.00	96.17	100.00	89.00	96.33	100.00
	LSTM	97.03	98.40	100.00	91.00	96.50	100.00	95.00	97.50	100.00	92.00	96.67	100.00
Spatial, temporal, posture, and orientation. Posture, orientation, and spatial determined using the proposed methods	HMM-F	85.84	92.88	96.48	87.00	93.17	96.00	86.00	92.17	96.00	85.00	92.33	96.00
	HMM-FB	86.14	93.01	96.48	87.00	93.00	96.00	88.00	92.83	96.00	86.00	92.67	96.00
	LSTM	87.86	93.32	96.51	86.00	91.50	96.00	87.00	92.33	96.00	86.00	92.17	96.00

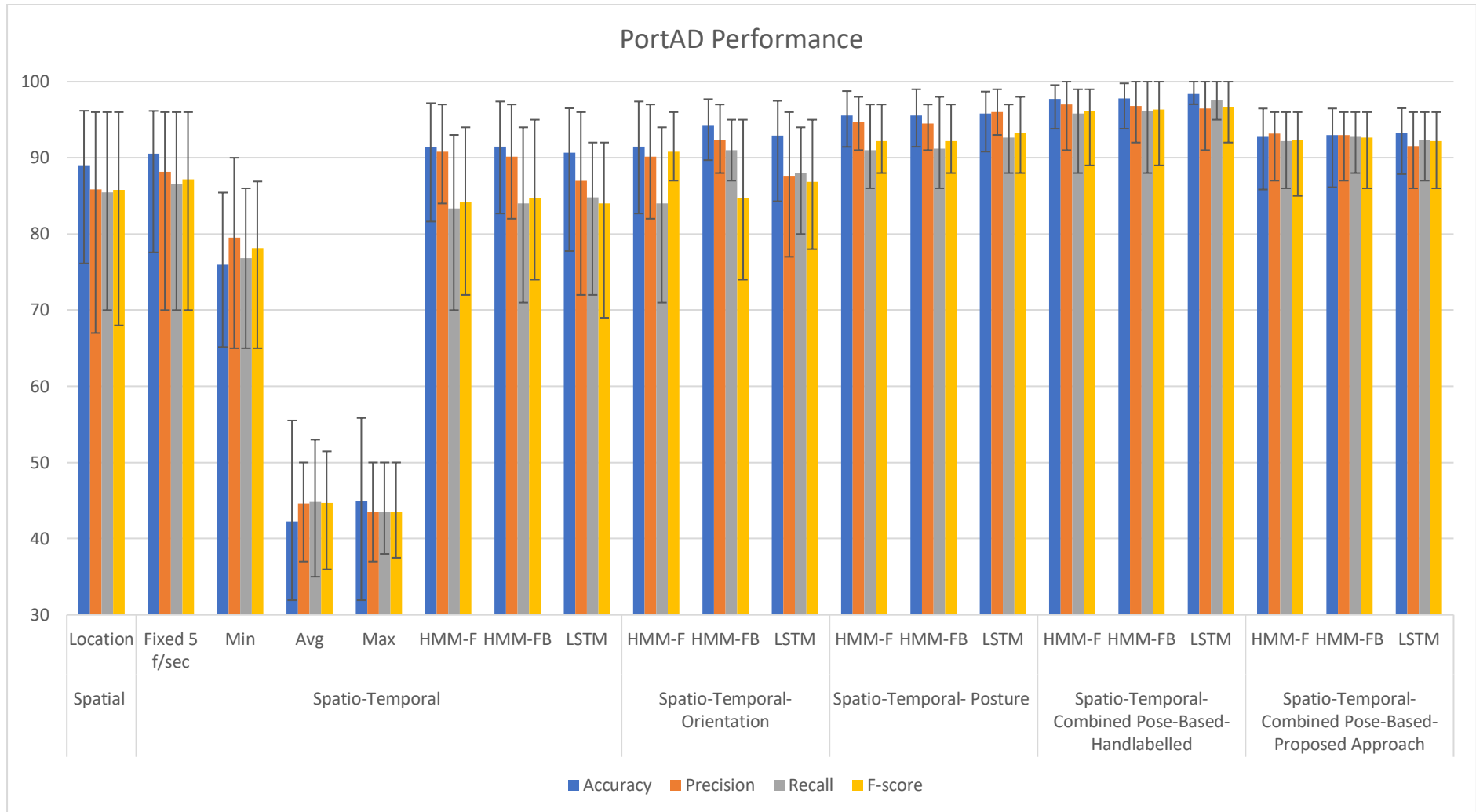


Figure 6. 2: Activity identification performance on the PortAD dataset (A-Min= Adaptive threshold Minimum, A-Max= Adaptive threshold Maximum, A-Avg= Adaptive threshold Average, HMM-F= Hidden Markov Model the Forward algorithm, HMM-FB =Hidden Markov Model the Forward-Backward algorithm). The main bars show the average values, and the error bars show the minimum and maximum values for each metric.

Table 6. 2: The performance values on data from the selected camera in the UR fall dataset using the proposed methods and features (HMM-F= Hidden Markov Model the Forward algorithm, HMM-FB =Hidden Markov Model the Forward-Backward algorithm). Represent the highest values for each feature combination. Represent the highest values across all feature combinations.

<i>Features</i>	<i>Method</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F-score</i>	<i>Specificity</i>
Spatial and temporal	HMM-F	54.7	58	61	59	77
	HMM-FB	61.25	64	64	68	78
	LSTM	64.8	46	64	54	77
Spatial, temporal and orientation. Orientation and spatial determined from true labels	HMM-F	94.34	92	97	94	100
	HMM-FB	94.38	92	97	94	100
	LSTM	93.7	91	96	93	100
Spatial, temporal, and posture. Posture and spatial determined from true labels	HMM-F	100	100	100	100	100
	HMM-FB	100	100	100	100	100
	LSTM	100	100	100	100	100
Spatial, temporal, posture and orientation. Posture, orientation, and spatial determined from true labels	HMM-F	100	100	100	100	100
	HMM-FB	100	100	100	100	100
	LSTM	100	100	100	100	100
Spatial, temporal, posture, and orientation. Posture, orientation, and spatial determined using the proposed methods	HMM-F	84.9	86	86	85	77.3
	HMM-FB	87.5	89	89	87	77.3
	LSTM	87.7	89	88	88	77.7

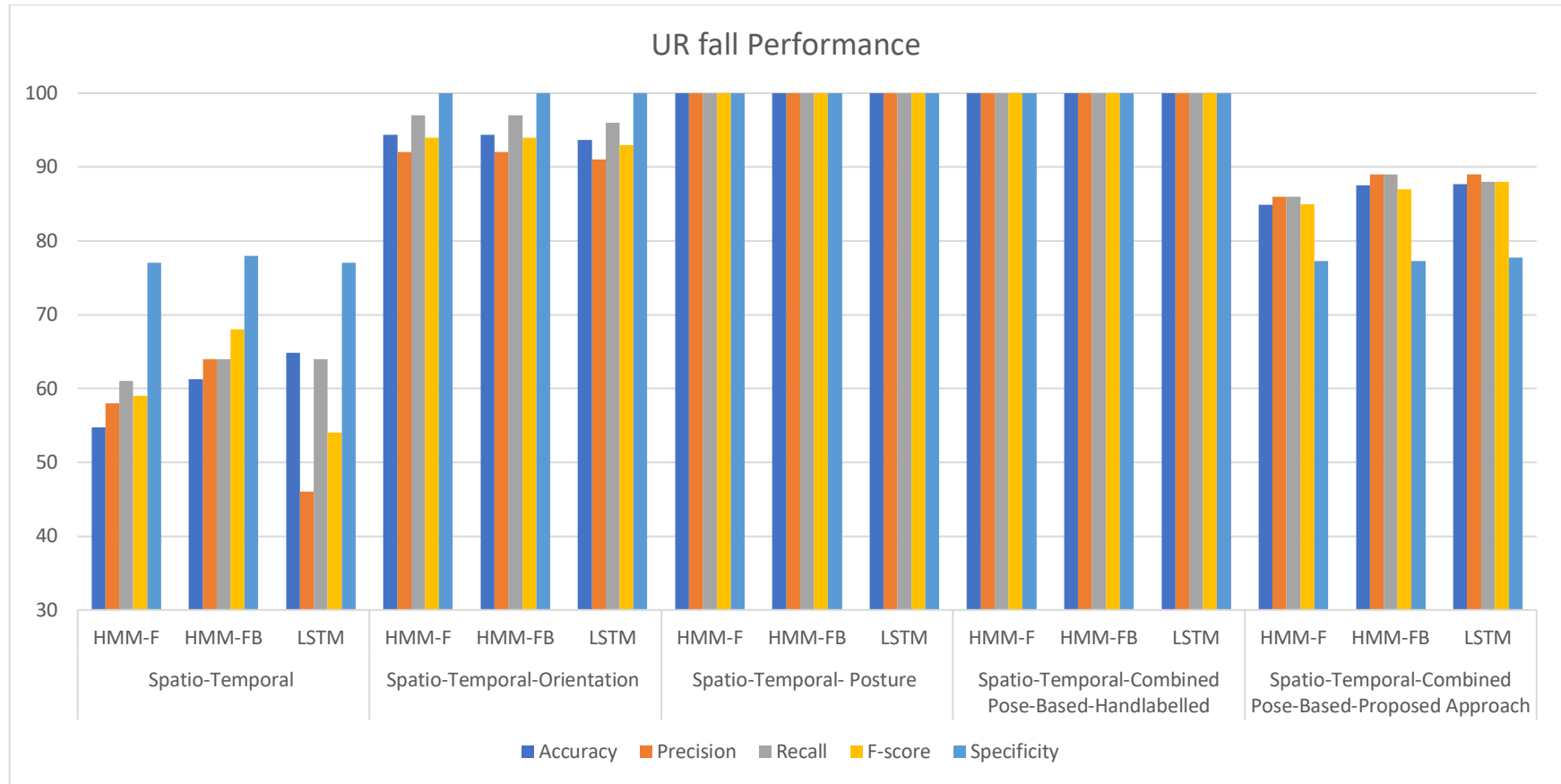


Figure 6. 3: The fall detection performance on the UR fall dataset using the proposed methods and features (HMM-F= Hidden Markov Model the Forward algorithm, HMM-FB =Hidden Markov Model the Forward-Backward algorithm)

Table 6. 3: The accuracy of detecting posture and orientation on the PortAD dataset ('Min', 'Max', and 'Avg' correspond to the minimum, maximum, and average performance across different cameras). Represent the highest values for each feature combination. Represent the highest values across all feature combinations.

Classifier →		Naive-Bayes			SVM			MLP			Decision Tree			Random Forest		
Task	Method	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
Orientation	Pairwise Distance	54.1	78.5	94.1	51.8	78.9	93.9	56.8	82.4	95.8	51.4	76.2	93.0	56.0	83.8	96.9
	Normalised Coordinates	53.9	82.8	97.3	53.9	86.0	97.7	58.3	86.9	97.4	59.3	85.9	97.2	55.1	86.9	98.2
Posture	Pairwise Distance	65.7	80.0	100.0	73.4	90.5	100.0	82.3	92.7	100.0	72.4	88.0	100.0	81.6	93.2	100.0
	Normalised Coordinates	73.0	85.8	100.0	74.1	90.5	100.0	85.1	93.4	100.0	71.9	90.4	100.0	84.3	93.7	100.0

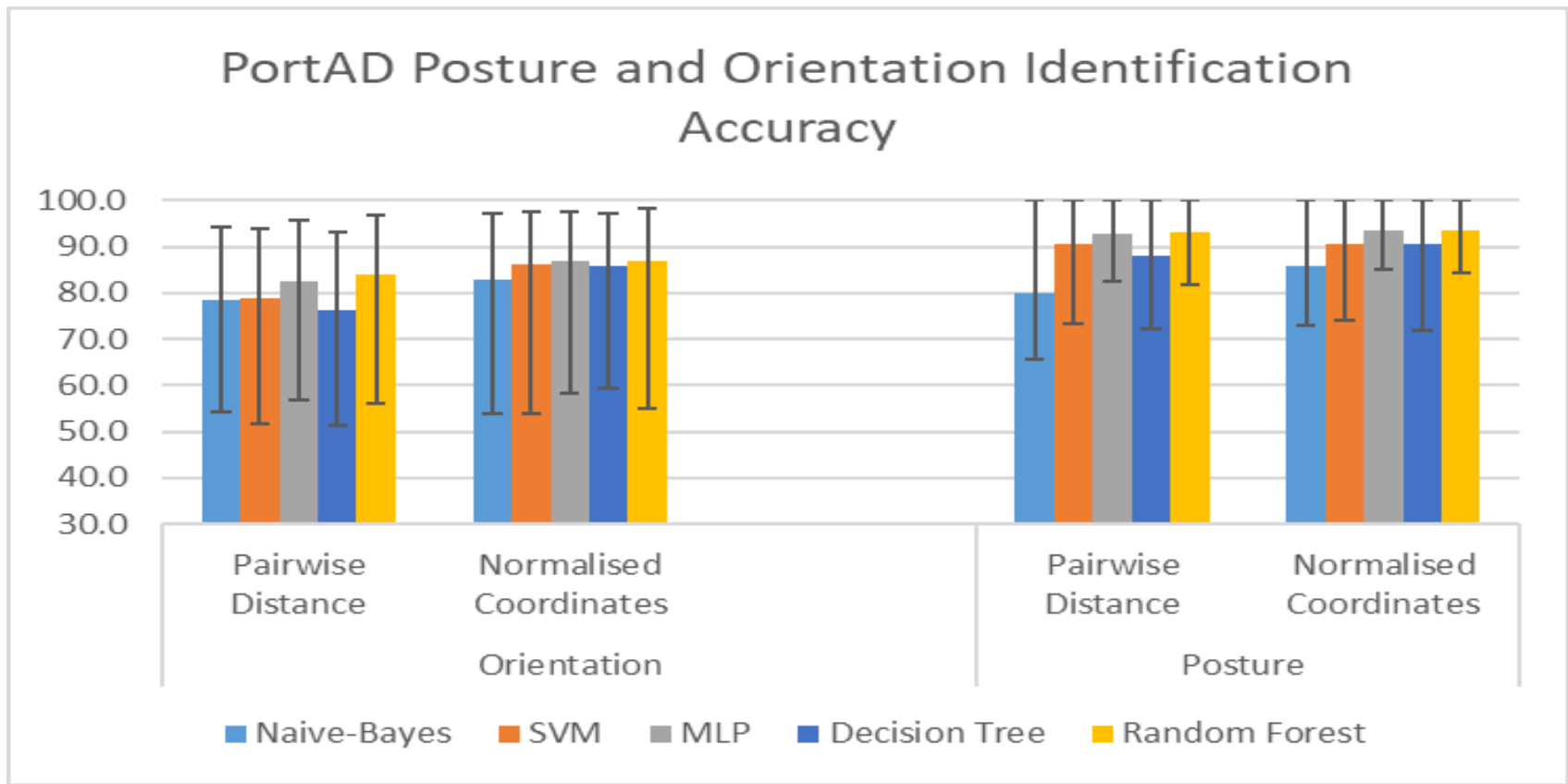


Figure 6. 4: The accuracy of detecting posture and orientation on the PortAD dataset. The main bars show the average value, and the error bars show the minimum and maximum values for each case.

Table 6. 4: The accuracy of detecting posture and orientation on the UR fall dataset. ■ Represent the highest values for each feature combination. ■ Represent the highest values across all feature combinations.

Task	Method	Naive-Bayes	SVM	MLP	Decision Tree	Random Forest
Orientation	Pairwise Distance	71.03	87.85	71.03	78.50	86.92
	Normalised Coordinates	78.50	81.31	85.98	92.52	88.79
Posture	Pairwise Distance	66.36	77.57	71.96	71.03	80.37
	Normalised Coordinates	60.75	80.37	78.50	81.31	85.98

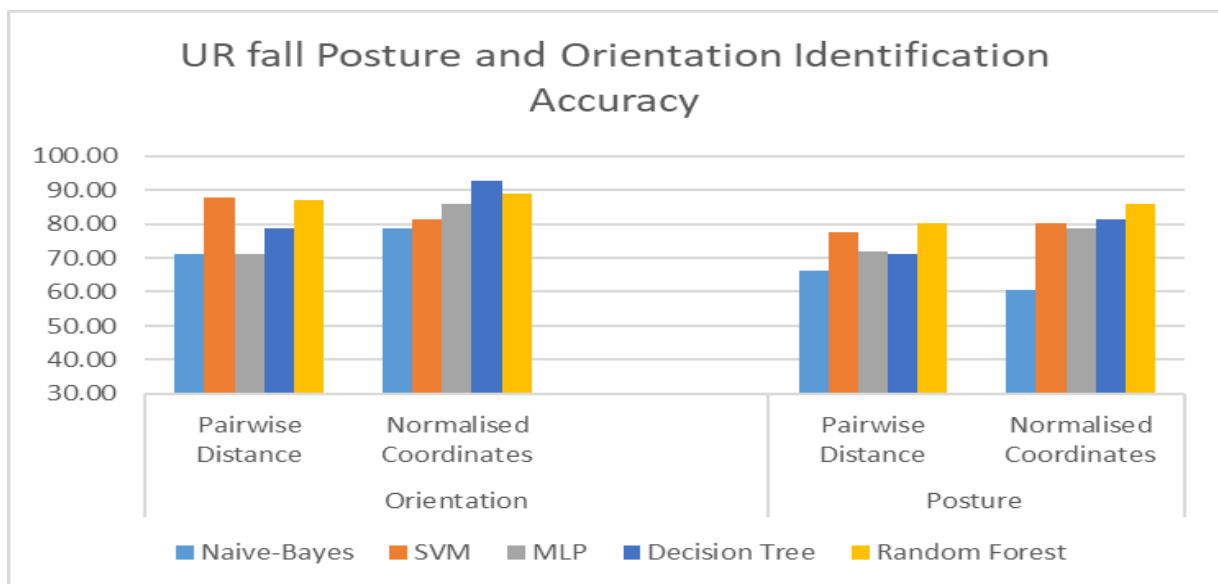


Figure 6. 5: The accuracy of detecting posture and orientation on the UR fall dataset

Table 6. 5: Pearson correlation coefficient values for the three conditional probability assumptions used to estimate emission probabilities in HMM when using all four high-level features.

	Minimum	Maximum
Assumption 1	0.65	0.93
Assumption 2	0.95	1
Assumption 3	0.95	1

6.7 Analysis

Using the location feature alone achieved good results when tested on the PortAD data. This shows that many high-level activities are very much location-specific and only a limited number of activities share the same location.

It can be seen from Figure 6. 2 that there is a big variation between the minimum and maximum activity identification performance for each method and feature combination. This happened because two cameras in the kitchen captured activities that share the same location. These activities are eating and preparing a meal. In the PortAD dataset, the labelled location is “dining table”, and the activities are eating or preparing a meal. Based on the spatial feature alone, only eating is identified as an activity. This has led to a 12% drop in the performance on the data from the two of the kitchen cameras covering the “dining table” location. These problems are expected, as the location feature on its own is not reliable enough to differentiate such activities.

Although the results using the location feature alone are good on the PortAD, this method cannot identify activities that share the same location, which limits its usefulness. Another critical issue with the location method is that it cannot identify any pose-based activities, such as a fall. That is why the location method has not been used to identify falls, as this method cannot identify activities that share the same location, and fall is a type of activity that can happen in all locations. This method does not require training data, which makes it simpler. However, this limits its usage and expandability.

Using location combined with a fixed time threshold achieved the best performance when the time threshold was equal to 5 frames. However, this performance is very close to the results using location only, with a minor improvement in performance by 1.5%. This performance improvement matched the expectations, as presented in Section 4. 1.

The activity-specific time threshold approach achieved lower performance for all of its variations, including using the minimum, maximum, and average activity durations as the activity-specific threshold values. The best performance for the activity-specific threshold approach was achieved when the minimum activity duration (as determined from the training data) was used as the threshold value. However, this was still lower than the performance for the fixed time threshold.

The fixed and adaptive time thresholds essentially solve only one additional problem compared to using the location feature alone. The time feature is used to differentiate between engaging in the activity at a particular location and just passing by. “A new activity is detected only when the person spends at a new location longer than the time threshold.” (Al-Wattar et al., 2016). However, that may also mean that short activities can be overlooked.

The Forward and the Forward-Backward algorithms achieved better performance compared to the time threshold methods. The machine learning algorithms overcome some of the problems that the fixed and adaptive time thresholds suffer from, that is identifying some of the activities that share the same location. The Forward-Backward algorithm achieved better performance compared to the Forward algorithm. This is expected as the Forward-Backward algorithm benefits from the additional information during the backward pass to help further reduce the uncertainty about the detected activity. The LSTM model achieved a slightly lower performance compared to the HMM models as can be seen from the results. This may be due to the insufficient amount of training data. The performance might increase and surpass the HMM when the training data size increases (Soekhoe, Van Der Putten & Plaat, 2016; Barbedo, 2018; Linjordet & Balog, 2019). Although the spatio-temporal methods identified some of the activities that share the same location, the achieved performance was low.

The performance for identifying fall from the spatial and temporal features using the proposed machine learning algorithms is very low, as the spatial and temporal features do not describe the sudden activity, i.e. change in the detected person posture.

The addition of the orientation feature improves the results compared to using only the spatial and temporal features. The orientation feature helps in distinguishing between activities that share the same location, because the person may have different orientations for different activities. For example, the spatial and temporal features would struggle in identifying the activities shown in Figure 6. 6. Adding orientation can provide a clear differentiation and improve performance.



Figure 6. 6: The reason for improving the performance when adding the orientation. The labelled activity for the left picture is using the sink, and the labelled activity for the right picture is walking. The two activities share the same location and the same posture.

Orientation is also helpful in identifying falls, as shown in the previous section. The reason for this is because the orientation as the extra feature improved the confidence level in identifying falls.

Adding the posture feature helps in improving the results by addressing the problem of activities sharing the same location and orientation but having different postures. An example can be seen in Figure 6. 7.

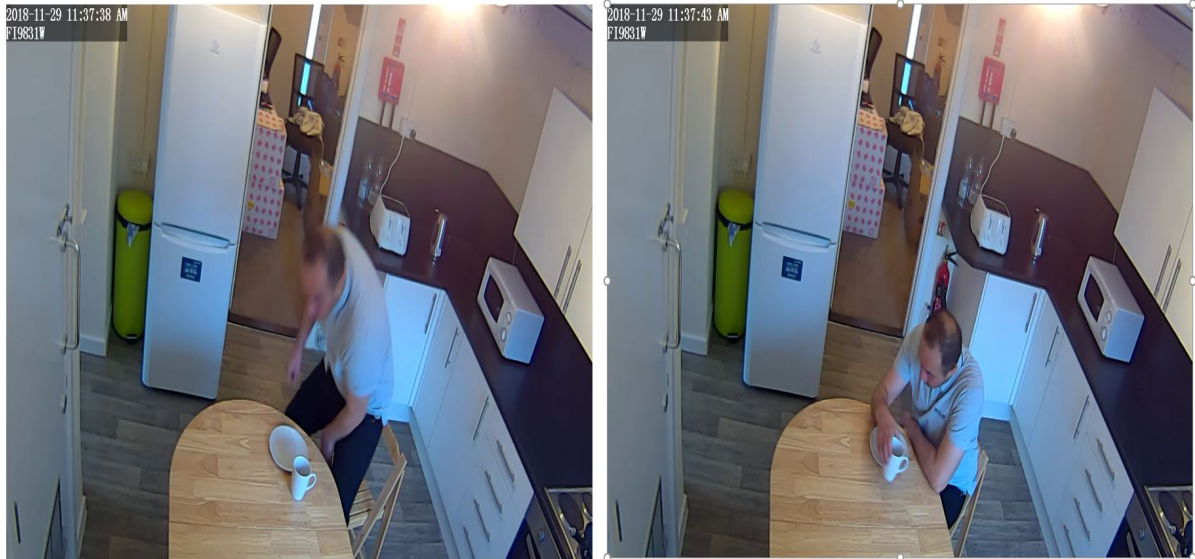


Figure 6. 7: The reason for improving the performance when adding the posture: the labelled activity for the left picture is preparing a meal (it is part of 'preparing a meal' activity), and the labelled activity for the right picture is eating. Two activities share the same location and same orientation

The results for using the posture as the third feature on the UR fall dataset were excellent. The reason for this is because the body shape is a very distinct characteristic during and after a fall (for instance, bending followed by lying).

Although using spatial and temporal features together with either the orientation or the posture improves the performance, there are some activities that share the same location and orientation with different posture as can be seen in Figure 6. 7, and others share the same location and the same posture with a different orientation, as shown in Figure 6. 6. Such activities cause the difference between the maximum and minimum values in the performance, shown in Section 6.6. The size of the dataset affects the performance of the methods. Therefore, increasing the size of the dataset might affect the performance and reduce the gap between the performance for the posture and orientation as a third feature.

Combining posture, orientation, location, and time features using HMM and LSTM achieved the best results. Combining the four features overcomes the problems that happen when using two or three features. In addition to that, combining the four features maintained the optimum performance for identifying falls, as it achieved the same results as the posture location and time features when tested on the UR fall dataset.

As suggested in Section 6.1, the activity identification can be affected not only by the applied algorithm (i.e. threshold based, HMM, or LSTM), but also by the accuracy of detecting the

features used for activity identification. The activity identification performance using the manually labelled features is higher compared to when the features are detected from the video data. This is because the errors in detecting features from video data can result in incorrect subsequent activity identification.

As explained in Chapter 4, the location is detected using the head coordinates obtained from the CPM. When the CPM does not detect a person in the video, the previous person's location, posture, and orientation are used. The percentage of instances using the previous person's location, posture, and orientation in the PortAD dataset is 8% of the total number of instances. Some of these instances may not have the features equal to the previous instance in the video sequence, resulting in feature detection errors, which may in turn lead to errors in activity identification. The BS-KNN performance for identifying the person's location is very low. The main cause for errors is the incorrect identification of the BS-KNN contour, as shown in Figure 6. 8.

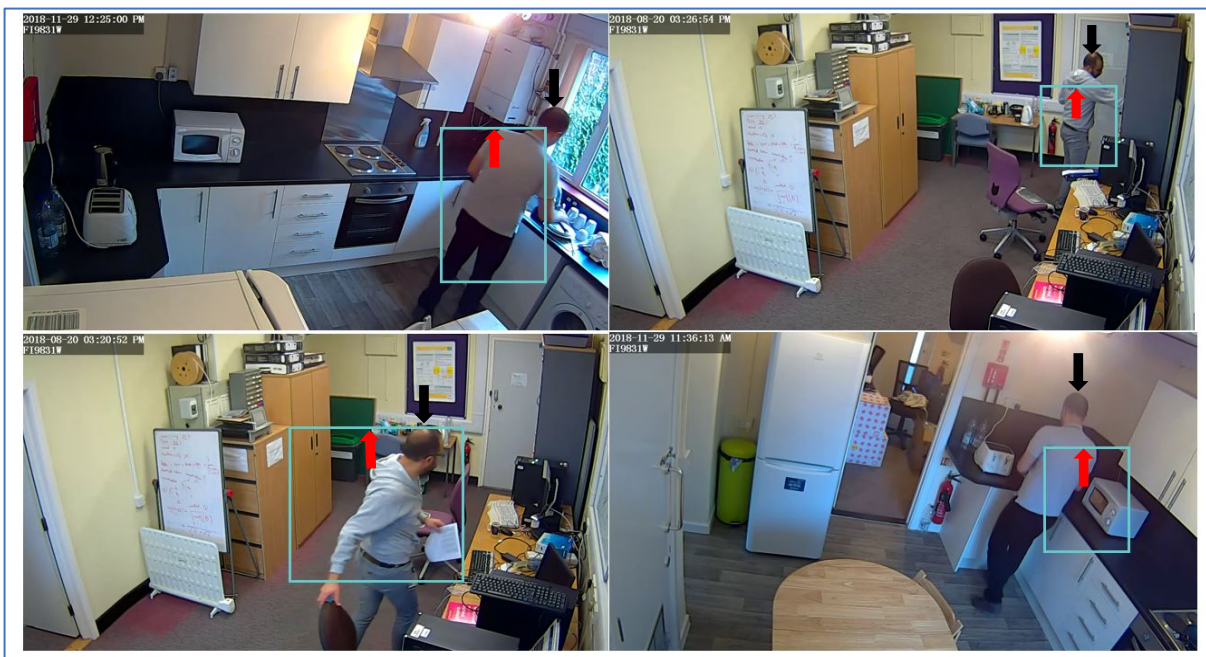


Figure 6. 8: Errors in location detection using the BS-KNN: the black arrows show the correct location (manually labelled), while the red arrows show the location detected with BS-KNN

Two approaches have been compared for posture and orientation detection. The reason that the classification with pairwise distances between body parts did not achieve the same performance as the classification with normalised body part coordinates is the type and the number of features. In the classification with pairwise distance, there are 91 features, some

of these features do not carry the necessary information, which makes the classification harder, and might reduce the performance. This might be the reason for the low performance in the classification with the pairwise method.

Classification with normalised body parts coordinates using the random forest classifier achieved the best results for the detection of posture and orientation on the PortAD dataset. Therefore, it is used in this work as the main feature extractor for the posture and orientation. The MLP classifier achieved a slightly lower performance, as shown in Section 6.6. On the UR fall dataset, the random forest classifier achieved the best performance on identifying the posture while the decision trees achieved the best performance on identifying the orientation. Therefore, the random forest classifier was selected as it achieved the highest classification performance on the majority of the tests.

There are some errors in identifying the posture and orientation. The errors can be seen in Figure 6. 10. In Figure 6. 10 (1), the orientation for the detected person can be away from the camera and right from the camera. It is labelled as right, but the method detected it as away. In Figure 6. 10 (2), the orientation is labelled as left, but the method detected it as facing. The posture in Figure 6. 10 (3) was detected as bending, while it is labelled as standing. For the same image, although the orientation can be right of the camera and facing, it was correctly identified as facing. Another problem is shown in Figure 6. 10 (4), where the CPM did not detect the person. Therefore, the previous state is used to identify the features. The posture for the detected person is taken from the previous frame, which is standing. Thus, the activity is more likely to be walking than using the phone (standing, facing the phone, next to the phone or in the walking area). In Figure 6. 10 (5), the orientation is labelled as facing, but it is detected as right of the camera. In Figure 6. 10 (6), the posture is labelled as bending, but it is detected as standing. Some of these errors happened because orientations can be labelled as both, such as in Figure 6. 10 (2, 3, & 5), where the subject was in a state of changing from one orientation to the other. Other errors happened because of posture identification, such as in Figure 6. 10 (6), where the error happened because the subject was partially occluded (the dining table hides the lower body). The CPM predicts the location for the lower body, and it is not the correct location. This leads to the wrong posture identification.



Figure 6. 9: Some of the problems in posture and orientation detection

The proposed approach for identifying the posture and orientation can be improved. Using multiple cameras to cover the same location from different angles, and selecting the cameras that achieve the best performance for identifying the posture and orientation has the potential to improve performance. For instance, one camera can be used for posture identification, and another one can be used simultaneously for orientation. The readings from the different cameras can then be combined. This can be implemented in future work.

Combining all four high-level features using HMMs relied on using conditional independence assumptions about posture, orientation, and location, as explained in Chapter 4. The experiments in this Chapter show that all three assumptions in Section 4.4 have a reasonable level of support in the PortAD dataset. In particular, Assumption 2 and Assumption 3 achieved

a strong linear relationship with the Pearson correlation above 0.7. Assumption 1 achieved a moderate positive relationship. According to the best practice in statistics (Moore & Kirkland, 2007; Mindrila & Balentyne, 2017), the three assumptions have moderate to strong levels of support.

6.8 Comparison of Feature Performance and Limitations

The experiments on the PortAD and the UR fall datasets show that the performance improved with using more high-level features. Using all four features (location, duration, posture, and orientation) achieves the best performance. The system managed to identify falls with high performance as well.

The machine learning models (HMM and LSTM) achieved very good results. LSTM achieved the best performance when the number of features increased.

A comparison between the selected features is presented in Table 6. 6.

Table 6. 6: A comparison between the selected features

Features and method	Advantages	Problems
Location	(1) High accuracy on PortAD dataset. (2) Very simple method.	(1) Cannot detect falls. (2) Cannot identify overlapped activities (activities that share the same location). (3) Hard to expand the model as it cannot learn automatically from training data.
Location and duration; Fixed time threshold	(1) Better accuracy compared to the location only, when using the correct threshold. (2) Simple method.	(1) Cannot detect falls. (2) Hard to expand the model as it cannot identify overlapped activity (activities that share the same location). (3) Hard to expand the model as it cannot learn automatically from training data. (4) The correct threshold depends on the user.
Location and duration; Activity-specific threshold	(1) Simple method.	(1) Cannot detect falls. (2) Cannot identify overlapped activity (activities that share the same location).
Location and duration; Machine Learning approaches	(1) Higher accuracy compared to the previous methods on the PortAD dataset. (2) Identifies fall with low accuracy.	(1) Low accuracy in identifying overlapped activities (activities that share the same location). (2) The performance for identifying falls is low. (3) Low accuracy in identifying activities that depend on posture (4) Low accuracy in identifying activities that depend on orientation.
Location, duration, and orientation; Machine Learning approaches	(1) Higher accuracy compared to the previous methods. (2) Can identify falls. (3) High accuracy in identifying activities that have different orientations.	(1) Lower accuracy in identifying overlapped activities that have the same location and orientation. (2) The performance for identifying falls needs improvement.
Location, duration, posture; Machine Learning approaches	(1) Higher accuracy compared to the previous methods. (2) Identify falls. (3) High accuracy in Identifying activities that have different postures.	(1) Low accuracy in identifying overlapped activities that have the same location and posture.
Location, duration, posture, and orientation; Machine Learning approaches	(1) Higher accuracy (Achieved the best results) compared to the previous methods (2) Identifies falls. (3) High accuracy in identifying overlapped activities that have the same two features (i.e. the same location and orientation and the same location and posture).	

Chapter 7 Conclusion

7.1 Discussion of Contributions

This section concludes the thesis by outlining the main findings of this work and presenting some recommendations for future development. The first main contribution of the study is the creation of a method for identifying the high-level activities of daily living, using high-level features that are the posture, orientation, location and time. The second main contribution is the creation of a dataset for evaluating the high-level activity identification achieved by the proposed method. The third main contribution of this study is addressing the effectiveness of the selected four high-level features in identifying activities of daily living.

The proposed model uses four high-level features – the location, activity duration (time), posture, and orientation of the detected person – to identify activity. The proposed approach uses three different machine learning algorithms: CNN, HMM or LSTM, and random forest classifier. Combined, these algorithms identify 14 high-level activity in two separate locations.

It was not possible to find an existing dataset that covered all the activities that this study targeted; hence, a new dataset was created. The Portsmouth Activity Dataset (PortAD) is an activity dataset that was recorded using multiple cameras in different indoor locations. In PortAD, 14 high-level activities were recorded, forming part of the activities of daily living (ADL) and instrumental activities of daily living (IADL). These ADL and IADL were performed in two locations and included four orientations, three postures, 14 locations, and 14 activities; the labelling for PortAD was based on all of these. The cameras were installed in the best location according to security experts, namely the top corners of the rooms; the reason for this was to provide better viewing angles and reduce occlusion by objects and people.

In order to assess the effectiveness of the features, the high-level features were tested using the hand-labelled data. Starting with the spatial feature alone, the average accuracy achieved was 89%. The spatial feature alone could not identify any activities sharing the same location, which made it impractical for identifying falls. The second test used spatial and temporal features; different methods were used for this: the fixed and adaptive time algorithms, the HMM and the LSTM. The fixed temporal threshold showed a slight improvement in performance of 1.5% compared to the location method when the best threshold was

selected. When the system was tested using the adaptive time thresholds, there was a decrease in performance for all methods; the details of this were provided in Chapter 6.

On the other hand, the Forward algorithm, the Forward–Backward algorithm, and the LSTM achieved average accuracies of 91.42%, 91.45%, and 90.68%, respectively. The machine learning algorithms identified fall activity but with a low performance; while the fixed temporal algorithms could not identify any sudden activity.

Next, three high-level features were tested. First, the spatial, temporal, and orientation features were combined. The average accuracies achieved using the Forward algorithm, the Forward-Backward algorithm, and the LSTM were 91.45%, 94.29%, and 92.89%, respectively. The three algorithms achieved very good performance when tested on the UR fall dataset, with accuracies of 94.34% for the Forward algorithm, 94.38% for the Forward–Backward algorithm, and 93.7% for the LSTM.

When the spatial, temporal, and posture features were combined, the average achieved accuracies using the Forward algorithm, the Forward–Backward algorithm, and the LSTM were 95.53%, 95.53%, and 95.82%, respectively. In PortAD, the posture feature combined with location and time achieved better performance compared to orientation combined with location and time. In addition, posture combined with location and time was able to detect sudden activity with 100% accuracy, a significant improvement compared to using fewer features.

The final test that was conducted combined of all four high-level features. The average achieved accuracies using the Forward algorithm, the Forward–Backward algorithm, and the LSTM were 97.74%, 97.82%, and 98.4%, respectively. In identifying fall, all the methods achieved 100% accuracy when tested on the UR fall dataset.

It can be concluded that using one feature to identify ADL is not sufficient, as it cannot identify activities in the same location or sudden activities. Different methods were used to identify ADL in this study, including time threshold and HMM with a classifier. Although it is easier to use time threshold, it is not recommended, as it achieved a lower performance compared to the other methods, and could not identify sudden activities or activities in the same location. Therefore, HMM and LSTM are used in the proposed system for combining the features.

Identifying activities from two features using HMM and LSTM improved the performance compared to using time threshold, and could identify both activities in the same location and sudden activities. However, the performance in identifying falls activities in the same location was low.

Using three features improved the performance compared to using two features, and successfully identified sudden activities with high performance. Adding posture as the third feature achieved a slightly better performance compared to orientation. However, a problem with using three features arose in relation to those activities that share two features. For example, when using posture as the third feature, it was observed that some activities shared the same location and posture, as presented in Chapter 6. Hence, it is recommended to use four features to avoid this issue; the fourth feature improves the confidence level for the method.

Combining the four high-level features and using LSTM and HMM achieved the best results in identifying the activities of daily living and falls. Based on the performance of this model, it can be said that increasing the number of high-level features improves the performance of activity identification.

When the proposed system combining multiple models of machine learning was used to extract the high-level features and identify activities, the achieved accuracies using the Forward algorithm, the Forward–Backward algorithm, and the LSTM were 92.88%, 93.01%, and 93.32%, respectively. The main reason for the deterioration in performance compared to the hand-labelled features was that the system’s performance in identifying posture and orientation was not ideal.

Overall, the proposed system achieved good performance. However, there remain a number of issues that need to be addressed. The first is the complexity of the system, being a combination of CNN, HMM, and LSTM. The second problem is the accumulated error, which is increased by the different levels and steps of training. These training levels are: the CNN, the classification for pose identification, and HMM/LSTM. Another issue with the proposed system is that it only identifies a single person’s activity.

Some areas in PortAD can be improved and added to in the future. In PortAD, one subject performed all the activities, a limitation which needs to be overcome in the future. The second

issue with PortAD is the limited locations, as it only covers two locations in the house. Also, no sudden activities were recorded in the dataset, and no activities were recorded at night using night mode and IR sensors. Other activities can be added to the dataset, for example group activities and shared location activities, such as two or more people doing similar or different activities in the same location.

A brief summary of the main contributions made in this work are as follows:

- Creating a method for identifying the high-level activities of daily living, using high-level features.
- Creating a dataset for evaluating high-level activity identification methods.
- Establishing the effectiveness of the selected four high-level features in identifying activities of daily living.
- Developing an algorithm to identify the activities of daily living in an indoor environment using four high-level features, namely: location, time, orientation, and posture.
- Testing the proposed method using a different number of features and different combinations to measure their effectiveness and impact on the results.
- Showing that the combination of the selected four high-level features and use of the proposed machine algorithms achieved the best results in identifying activities of daily living, including sudden activity.

7.2 Future Work

There are multiple directions and areas of improvement that could be explored in future studies, which can be summarised as follows:

- Attempts to improve the confidence level for the proposed methods and the performance of the proposed system and reduce the system's complexity.
- In addition, a future study could increase the size and the variety of the dataset by recording two or more subjects performing activities together, or group activities; i.e., subjects doing the same activity, and performing different activities in the same location, for example Subject 1 working on a computer while Subject 2 is on the phone. Adding more indoor test areas in which to perform the activities, and performing activities at night using

the cameras' night mode. Other activities and subjects could be added to the dataset, such as falls when sleeping and falling on stairs, and pets.

- The present study showed that increasing the number of features improved the system performance. Therefore, a future study could investigate the effectiveness of adding more features, such as voice, time of day, day of the week, month, season, temperature, humidity, and light. Although some of these features are ambient features, it is nonetheless worth testing to see whether they affect the performance of the system.
- Identify both low-level and high-level activities, by using low-level features. The proposed method identifies high-level activities only. To identify a wider range of activities, it is recommended to address the effectiveness of adding low-level features into the system. This could be achieved using methods such as dense trajectory or optical flow with Histogram with Oriented Gradient (HoG), or Haar-like feature descriptors (Viola & Jones, 2001). These features may improve the system's detection and help in identifying low-level activities in addition to high-level activities.
- The proposed approach method with images and cannot work with a frame rate of 30 frames per second or higher. The reason for that is the used CPM, which can only read nine images per second. Therefore, the CPM could be replaced with that used by Cao et al. (2017), namely a 2D pose estimation using part affinity fields; according to their results, this works successfully with a frame rate of 30 frames per second. It would be informative and worthwhile to test the system's performance in identifying posture and orientation using this approach.
- Identifying group activities that can be performed by two or more subjects and identifying the activities of more than one person in the same location, so that the system can be used in care homes and special schools. The selected CNN model cannot detect multiple people at the same time when they are partially occluded. Thus, it is recommended to use a different model, such as the 2D pose estimation using part affinity fields, or to develop a new model based on a new dataset to cover multiple subjects.
- The HMM and LSTM achieved good performance when tested in the proposed system. Therefore, it is suggested that different machine learning algorithms be tested to improve on this performance, such as Conditional Random Fields (CRF) (Lafferty, McCallum & Pereira, 2001), or Hidden Conditional Random Fields (Quattoni, Wang, Morency, Collins &

Darrell, 2007), which have been used in gesture recognition and might outperform the HMM and CRF (Wang, Quattoni, Morency, Demirdjian & Darrell, 2006). Future researchers could also try different models of LSTM, such as bidirectional LSTM (BiLSTM) (Huang, Xu & Yu, 2015; Chen, Xu, He & Wang, 2017), and try different types of sliding windows.

- Improve the privacy of the system by adding extra encryption keys to a program located in the camera. One feasible idea is to program and install a Raspberry Pi or an Arduino system and use these to encrypt the data from the camera. Alternatively, the code can be installed on a card next to the camera; the camera can identify the activity, and only the output data taken from the camera will be encrypted.
- In order to create a complete system that can help the monitored people, an action suggestion or actuation is required. Future researchers could attempt to create a system that can identify activity, send text messages and emails, start a video call, or call for an emergency, to help the monitored person live independently.
- In this work, the researcher had to label all the locations in the recorded rooms. To make the system more practical and applicable in wider locations, automatic labelling systems for the locations based on objects is required. Therefore, it is suggested that future researchers use a method to identify the objects in the room and determine the exact location of the detected person based on the distance between an object and the person. Indeed, work on this has already commenced. First, ResNet (He et al., 2016) was trained on COCO dataset (Lin et al., 2014), and used as an object identifier in the room, but the model did not perform well, and could only identify few objects. Then YOLO9000 (Redmon & Farhadi, 2017) was trained on COCO dataset and was able to identify more objects than ResNet in the kitchen and the home office. The performance for YOLO9000 requires improvement, as, while it identified all the objects in the location, there were errors as some of the objects were wrongly identified, as shown in Figure 7. 1. The aim is to filter the items that are likely to be in the location and, based on that, measure the distance of the detected person from the object and use this information for location identification.



Figure 7. 1 Object detection using CNN (YOLO9000)

References

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Ghemawat, S. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.
- Adami, A. M., Hayes, T. L., & Pavel, M. (2003, September). Unobtrusive monitoring of sleep patterns. In Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE (Vol. 2, pp. 1360-1363). IEEE.
- Aggarwal, J. K., & Ryoo, M. S. (2011). Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 43(3), 16.
- Ahlgren, P., Jarneving, B., & Rousseau, R. (2003). Requirements for a cocitation similarity measure, with special reference to Pearson's correlation coefficient. *Journal of the American Society for Information Science and Technology*, 54(6), 550-560.
- Al-Wattar, M., Khusainov, R., Azzi, D., & Chiverton, J. (2016, October). Activity recognition from video data using spatial and temporal features. In 12th International Conference on Intelligent Environments. IEEE.
- Anderson, C. A., & Bushman, B. J. (2002). Human aggression. *Annual review of psychology*, 53.
- Anderson, J. A., & Hinton, G. E. (2014). Models of information processing in the brain. In *Parallel models of associative memory* (pp. 33-74). Psychology Press.
- Andriluka, M., Pishchulin, L., Gehler, P., & Schiele, B. (2014). 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition* (pp. 3686-3693).
- Andriluka, M., Roth, S., & Schiele, B. (2009, June). Pictorial structures revisited: People detection and articulated pose estimation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (pp. 1014-1021). IEEE.

Arnold, L. M., Ball, M. J., Duncan, A. W., & Mann, J. (1993). Effect of isoenergetic intake of three or nine meals on plasma lipoproteins and glucose metabolism. *The American journal of clinical nutrition*, 57(3), 446-451.

Asada, H. H., Shaltis, P., Reisner, A., Rhee, S., & Hutchinson, R. C. (2003). Mobile monitoring with wearable photoplethysmographic biosensors. *IEEE engineering in medicine and biology magazine*, 22(3), 28-40.

ash, A. (2016). Population dynamics of UK city regions since mid-2011.

Auvinet, E., Rougier, C., Meunier, J., St-Arnaud, A., & Rousseau, J. (2010). Multiple cameras fall dataset. DIRO-Université de Montréal, Tech. Rep, 1350.

Aziz, O., Robinovitch, S. N., & Park, E. J. (2016, August). Identifying the number and location of body worn sensors to accurately classify walking, transferring and sedentary activities. In *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the* (pp. 5003-5006). IEEE.

Aztiria, A., Augusto, J. C., Izaguirre, A., & Cook, D. (2009). Learning accurate temporal relations from user actions in intelligent environments. In *3rd Symposium of Ubiquitous Computing and Ambient Intelligence 2008* (pp. 274-283). Springer, Berlin, Heidelberg.

Ba, J., & Caruana, R. (2014). Do deep nets really need to be deep?. In *Advances in neural information processing systems*(pp. 2654-2662).

Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., & Baskurt, A. (2011, November). Sequential deep learning for human action recognition. In *International workshop on human behavior understanding* (pp. 29-39). Springer, Berlin, Heidelberg.

Bahadori, S., Grisetti, G., Iocchi, L., Leone, G. R., & Nardi, D. (2005, June). Real-time tracking of multiple people through stereo vision. In *Proc. of IEE international workshop on intelligent environments*.

Baldi, P., & Chauvin, Y. (1996). Hybrid modeling, HMM/NN architectures, and protein applications. *Neural Computation*, 8(7), 1541-1565.

Banerjee, T., Keller, J. M., Skubic, M., & Stone, E. (2014). Day or night activity recognition from video using fuzzy clustering techniques. *IEEE Transactions on Fuzzy Systems*, 22(3), 483-493.

Bansal, S., Khandelwal, S., Gupta, S., & Goyal, D. (2013, September). Kitchen activity recognition based on scene context. In 2013 IEEE International Conference on Image Processing (pp. 3461-3465). IEEE.

Barbedo, J. G. A. (2018). Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. *Computers and electronics in agriculture*, 153, 46-53.

Barnich, O., & Van Droogenbroeck, M. (2010). ViBe: A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image processing*, 20(6), 1709-1724.

Bengio, Y., LeCun, Y., Nohl, C., & Burges, C. (1995). LeRec: A NN/HMM hybrid for on-line handwriting recognition. *Neural Computation*, 7(6), 1289-1303.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2), 157-166.

Berthold, M. R., Cebon, N., Dill, F., Gabriel, T. R., Kötter, T., Meinel, T., ... & Wiswedel, B. (2009). KNIME-the Konstanz information miner: version 2.0 and beyond. *AcM SIGKDD explorations Newsletter*, 11(1), 26-31.

Bikel, D. M., Schwartz, R., & Weischedel, R. M. (1999). An algorithm that learns what's in a name. *Machine learning*, 34(1-3), 211-231.

Bilen, H., Fernando, B., Gavves, E., & Vedaldi, A. (2017). Action recognition with dynamic image networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Biliński, P. T. (2014). Human action recognition in videos (Doctoral dissertation, Université Nice Sophia Antipolis).

Bisong, E. (2019). Google Colaboratory. In *Building Machine Learning and Deep Learning Models on Google Cloud Platform* (pp. 59-64). Apress, Berkeley, CA.

Bloom, V., Makris, D., & Argyriou, V. (2012, June). G3d: A gaming action dataset and real time action recognition evaluation framework. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on* (pp. 7-12). IEEE.

Bradford, W. C. (2004). Reaching the visual learner: teaching property through art. *The Law Teacher*, 11.

Brand, M., Oliver, N., & Pentland, A. (1997, June). Coupled hidden Markov models for complex action recognition. In *Computer vision and pattern recognition, 1997. proceedings., 1997 IEEE computer society conference on* (pp. 994-999). IEEE.

Brodie, M. A., Wang, K., Delbaere, K., Persiani, M., Lovell, N. H., Redmond, S. J., ... & Lord, S. R. (2015). New methods to monitor stair ascents using a wearable pendant device reveal how behavior, fear, and frailty influence falls in octogenarians. *IEEE Transactions on Biomedical Engineering*, 62(11), 2595-2601.

Brownlee, J. (2018). *Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python. Machine Learning Mastery.*

Bulat, A., & Tzimiropoulos, G. (2016, October). Human pose estimation via convolutional part heatmap regression. In *European Conference on Computer Vision* (pp. 717-732). Springer, Cham.

Burenus, M., Sullivan, J., & Carlsson, S. (2013). 3D pictorial structures for multiple view articulated pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3618-3625).

Burleson, W., Lozano, C., Ravishankar, V., Lee, J., & Mahoney, D. (2018). An assistive technology system that provides personalized dressing support for people living with dementia: capability study. *JMIR medical informatics*, 6(2), e21.

Cao, S., & Nevatia, R. (2016, December). Exploring deep learning based solutions in fine grained activity recognition in the wild. In *Pattern Recognition (ICPR), 2016 23rd International Conference on* (pp. 384-389). IEEE.

Cao, Z., Simon, T., Wei, S. E., & Sheikh, Y. (2016). Realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1611.08050*.

Carneiro, T., Da Nóbrega, R. V. M., Nepomuceno, T., Bian, G. B., De Albuquerque, V. H. C., & Reboucas Filho, P. P. (2018). Performance analysis of google colab as a tool for accelerating deep learning applications. *IEEE Access*, 6, 61677-61685.

Chan, M., Estève, D., Escriba, C., & Campo, E. (2008). A review of smart homes—Present state and future challenges. *Computer methods and programs in biomedicine*, 91(1), 55-81.

Charfi, I., Miteran, J., Dubois, J., Atri, M., & Tourki, R. (2013). Optimized spatio-temporal descriptors for real-time fall detection: comparison of support vector machine and Adaboost-based classification. *Journal of Electronic Imaging*, 22(4), 041106.

Chen, B., Fan, Z., & Cao, F. (2015, July). Activity recognition based on streaming sensor data for assisted living in smart homes. In *Intelligent Environments (IE), 2015 International Conference on* (pp. 124-127). IEEE.

Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2018). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4), 834-848.

Chen, T., Xu, R., He, Y., & Wang, X. (2017). Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN. *Expert Systems with Applications*, 72, 221-230.

Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Choi, W., Shahid, K., & Savarese, S. (2011, June). Learning context for collective activity recognition. In *CVPR 2011* (pp. 3273-3280). IEEE.

Chollet, F., (2015). keras.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Cilla, R., Patricio, M. A., García, J., Berlanga, A., & Molina, J. M. (2009). Recognizing human activities from sensors using hidden markov models constructed by feature selection techniques. *Algorithms*, 2(1), 282-300.

Cippitelli, E., Gasparrini, S., Gambi, E., & Spinsante, S. (2016). A human activity recognition system using skeleton data from RGBD sensors. *Computational intelligence and neuroscience*, 2016, 21.

Cook, D. J., & Das, S. K. (2007). How smart are our environments? An updated look at the state of the art. *Pervasive and mobile computing*, 3(2), 53-73.

Cook, D., Schmitter-Edgecombe, M., Crandall, A., Sanders, C., & Thomas, B. (2009, April). Collecting and disseminating smart home sensor data in the CASAS project. In *Proceedings of*

the CHI workshop on developing shared home behavior datasets to advance HCI and ubiquitous computing research (pp. 1-7).

Cosi, P. (2000, July). Hybrid HMM-NN architectures for connected digit recognition. In *ijcnn* (p. 5085). IEEE.

Daher, M., Diab, A. E. B. E., El Najjar, M. E. B., Khalil, M., & Charpillat, F. (2016). Elder tracking and fall detection system using smart tiles. *Sensors*, 15800(1).

Dahl, G. E., Yu, D., Deng, L., & Acero, A. (2012). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on audio, speech, and language processing*, 20(1), 30-42.

Damen, D., Doughty, H., Farinella, G. M., Fidler, S., Furnari, A., Kazakos, E., ... & Wray, M. (2018). Scaling Egocentric Vision: The EPIC-KITCHENS Dataset. *arXiv preprint arXiv:1804.02748*.

Davis, J., & Goadrich, M. (2006, June). The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning* (pp. 233-240). ACM.

De Souza, C. R., Gaidon, A., Cabon, Y., & Peña, A. M. L. (2017, July). Procedural Generation of Videos to Train Deep Action Recognition Networks. In *CVPR* (pp. 2594-2604).

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (pp. 248-255). IEEE.

Dohr, A., Modre-Opstrian, R., Drobics, M., Hayn, D., & Schreier, G. (2010, April). The internet of things for ambient assisted living. In *2010 seventh international conference on information technology: new generations* (pp. 804-809). IEEE.

Dubois, A., & Charpillat, F. (2013, July). Human activities recognition with RGB-Depth camera using HMM. In *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE* (pp. 4666-4669). IEEE.

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(Jul), 2121-2159.

Duckworth, P., Al-Omari, M., Charles, J., Hogg, D. C., & Cohn, A. G. (2017, February). Latent Dirichlet Allocation for Unsupervised Activity Analysis on an Autonomous Mobile Robot. In *AAAI* (pp. 3819-3826).

Ejupi, A., Brodie, M., Lord, S. R., Annegarn, J., Redmond, S. J., & Delbaere, K. (2017). Wavelet-based sit-to-stand detection and assessment of fall risk in older people using a wearable pendant device. *IEEE Transactions on Biomedical Engineering*, 64(7), 1602-1607.

Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1), 98-136.

Fei, L. (2012, May). Combining pictorial structure and image features to estimate human pose. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on* (pp. 1764-1768). IEEE.

Feichtenhofer, C., Pinz, A., & Wildes, R. P. (2017, July). Spatiotemporal multiplier networks for video action recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 7445-7454). IEEE.

Fei-Fei, L., Fergus, R., & Perona, P. (2006). One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4), 594-611.

Felzenszwalb, P. F., & Huttenlocher, D. P. (2005). Pictorial structures for object recognition. *International journal of computer vision*, 61(1), 55-79.

Fennelly, L. (2016). *Effective physical security*. Butterworth-Heinemann.

Fernando, B., Anderson, P., Hutter, M., & Gould, S. (2016). Discriminative hierarchical rank pooling for activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1924-1932).

Ferrari, V., Marin-Jimenez, M., & Zisserman, A. (2009, June). Pose search: retrieving people using their pose. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1-8). IEEE.

Fiske, A. P. (1992). The four elementary forms of sociality: framework for a unified theory of social relations. *Psychological review*, 99(4), 689.

Fleck, S., & Straßer, W. (2008). Smart camera-based monitoring system and its application to assisted living. *Proceedings of the IEEE*, 96(10), 1698-1714.

Fleck, S., Busch, F., & Straßer, W. (2006). Adaptive probabilistic tracking embedded in smart cameras for distributed surveillance in a 3D model. *EURASIP Journal on Embedded Systems*, 2007(1), 029858.

Gaglio, S., Re, G. L., & Morana, M. (2015). Human Activity Recognition Process Using 3-D Posture Data. *IEEE Trans. Human-Machine Systems*, 45(5), 586-597.

Gaikwad, K. (2012). HMM classifier for human activity recognition. *Computer Science & Engineering*, 2(4), 27.

Gales, M. J. (1998). Maximum likelihood linear transformations for HMM-based speech recognition. *Computer speech & language*, 12(2), 75-98.

Gales, M., & Young, S. (2008). The application of hidden Markov models in speech recognition. *Foundations and Trends® in Signal Processing*, 1(3), 195-304.

Gers, F. A., & Schmidhuber, J. (2000, July). Recurrent nets that time and count. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium (Vol. 3, pp. 189-194)*. IEEE.

Gers, F. A., Schraudolph, N. N., & Schmidhuber, J. (2002). Learning precise timing with LSTM recurrent networks. *Journal of machine learning research*, 3(Aug), 115-143.

Ghosh, A., Chakraborty, D., Prasad, D., Saha, M., & Saha, S. (2018, January). Can we recognize multiple human group activities using ultrasonic sensors?. In *Communication Systems & Networks (COMSNETS), 2018 10th International Conference on (pp. 557-560)*. IEEE.

Gia, T. N., Tcareno, I., Sarker, V. K., Rahmani, A. M., Westerlund, T., Liljeberg, P., & Tenhunen, H. (2016, November). Iot-based fall detection system with energy efficient sensor nodes. In *Nordic Circuits and Systems Conference (NORCAS), 2016 IEEE (pp. 1-6)*. IEEE.

Gilks, W. R., Richardson, S., & Spiegelhalter, D. (1995). *Markov chain Monte Carlo in practice*. Chapman and Hall/CRC.

Glanz, K., Rimer, B. K., & Viswanath, K. (Eds.). (2008). Health behavior and health education: theory, research, and practice. John Wiley & Sons.

Glorot, X., & Bengio, Y. (2010, March). Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics (pp. 249-256).

Go, R., & Aoki, Y. (2016, October). Flexible top-view human pose estimation for detection system via CNN. In Consumer Electronics, 2016 IEEE 5th Global Conference on (pp. 1-4). IEEE.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., & Schmidhuber, J. (2008). A novel connectionist system for unconstrained handwriting recognition. IEEE transactions on pattern analysis and machine intelligence, 31(5), 855-868.

Graves, A., Mohamed, A. R., & Hinton, G. (2013, May). Speech recognition with deep recurrent neural networks. In 2013 IEEE international conference on acoustics, speech and signal processing (pp. 6645-6649). IEEE.

Guan, Y., & Plötz, T. (2017). Ensembles of deep lstm learners for activity recognition using wearables. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 1(2), 11.

Guesgen, H. W. (2015, July). Towards a Theory of Space for Activity Recognition in Smart Environments Based on Rough Sets. In Intelligent Environments (IE), 2015 International Conference on (pp. 148-151). IEEE.

Guo, F., He, Y., & Guan, L. (2017, November). RGB-D camera pose estimation using deep neural network. In Signal and Information Processing (GlobalSIP), 2017 IEEE Global Conference on (pp. 408-412). IEEE.

Gupta, P., & Dallas, T. (2014). Feature selection and activity recognition system using a single triaxial accelerometer. IEEE Transactions on Biomedical Engineering, 61(6), 1780-1786.

Han, K., Yu, D., & Tashev, I. (2014). Speech emotion recognition using deep neural network and extreme learning machine. In Fifteenth annual conference of the international speech communication association.

Hanai, Y., Nishimura, J., & Kuroda, T. (2009, January). Haar-like filtering for human activity recognition using 3d accelerometer. In Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop, 2009. DSP/SPE 2009. IEEE 13th (pp. 675-678). IEEE.

Handa, A., Whelan, T., McDonald, J., & Davison, A. J. (2014, May). A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In 2014 IEEE international conference on Robotics and automation (ICRA) (pp. 1524-1531). IEEE.

Haritaoglu, I., Harwood, D., & Davis, L. S. (2000). W4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (8), 809-830.

Harvey, N. M., Zhou, Z., Keller, J. M., Rantz, M. J., & He, Z. (2009). Automated estimation of elder activity levels from anonymized video data. *Electrical and Computer Engineering publications (MU)*.

Hauptmann, A. G., Gao, J., Yan, R., Qi, Y., Yang, J., & Wactlar, H. D. (2004). Automated analysis of nursing home observations. *IEEE Pervasive Computing*, (2), 15-21.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision* (pp. 1026-1034).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

Heaton, J. (2015). *AIFH, Volume 3: Deep Learning and Neural Networks*.

Helal, S., Mann, W., El-Zabadani, H., King, J., Kaddoura, Y., & Jansen, E. (2005). The gator tech smart house: A programmable pervasive space. *Computer*, 38(3), 50-60.

Helal, S., Winkler, B., Lee, C., Kaddoura, Y., Ran, L., Giraldo, C., ... & Mann, W. (2003, March). Enabling location-aware pervasive computing applications for the elderly. In *Pervasive Computing and Communications, 2003.(PerCom 2003). Proceedings of the First IEEE International Conference on* (pp. 531-536). IEEE.

Hinton, G., Srivastava, N., & Swersky, K. (2012). Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. Cited on, 14(8).

- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.
- Hu, J., Brown, M. K., & Turin, W. (1996). HMM based online handwriting recognition. *IEEE Transactions on pattern analysis and machine intelligence*, 18(10), 1039-1045.
- Huang, H. L., Shyu, Y. I. L., Chen, M. C., Huang, C. C., Kuo, H. C., Chen, S. T., & Hsu, W. C. (2015). Family caregivers' role implementation at different stages of dementia. *Clinical interventions in aging*, 10, 135.
- Huang, Y. C., Yi, C. W., Peng, W. C., Lin, H. C., & Huang, C. Y. (2017, August). A study on multiple wearable sensors for activity recognition. In *Dependable and Secure Computing, 2017 IEEE Conference on* (pp. 449-452). IEEE.
- Huang, Z., Wan, C., Probst, T., & Van Gool, L. (2017). Deep learning on lie groups for skeleton-based action recognition. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1243-1252). IEEE computer Society.
- Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*.
- Ibrahim, M. S., Muralidharan, S., Deng, Z., Vahdat, A., & Mori, G. (2016). A hierarchical deep temporal model for group activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1971-1980).
- Jalal, A., Kamal, S., & Kim, D. (2015, October). Individual detection-tracking-recognition using depth activity images. In *Ubiquitous Robots and Ambient Intelligence (URAI), 2015 12th International Conference on* (pp. 450-455). IEEE.
- Jammalamadaka, N., Zisserman, A., Eichner, M., Ferrari, V., & Jawahar, C. V. (2012, October). Has my algorithm succeeded? an evaluator for human pose estimators. In *European Conference on Computer Vision* (pp. 114-128). Springer, Berlin, Heidelberg.
- Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., ... & Darrell, T. (2014, November). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia* (pp. 675-678).

Johnson, S., & Everingham, M. (2010, August). Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation. In *bmvc* (Vol. 2, No. 4, p. 5).

Johnson, S., & Everingham, M. (2011, June). Learning effective human pose estimation from inaccurate annotation. In *Computer vision and pattern recognition (CVPR), 2011 IEEE conference on* (pp. 1465-1472). IEEE.

Joint Committee for Guides in Metrology. (2008). *International vocabulary of metrology—Basic and general concepts and associated terms (VIM)*.

Jurafsky, D., & Martin, J. H. (2014). *Speech and language processing* (Vol. 3). London:: Pearson.

Kabir, M. H., Hoque, M. R., Thapa, K., & Yang, S. H. (2016). Two-layer hidden Markov model for human activity recognition in home environments. *International Journal of Distributed Sensor Networks*, 12(1), 4560365.

Kadkhodamohammadi, A., Gangi, A., de Mathelin, M., & Padoy, N. (2017, March). A Multi-view RGB-D Approach for Human Pose Estimation in Operating Rooms. In *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on* (pp. 363-372). IEEE.

Kario, K., Yasui, N., & Yokoi, H. (2003). Ambulatory blood pressure monitoring for cardiovascular medicine. *IEEE Engineering in Medicine and Biology Magazine*, 22(3), 81-88.

Karpathy, A. (2015). The unreasonable effectiveness of recurrent neural networks. *Andrej Karpathy blog*, 21, 23.

Ke, Q., Bennamoun, M., An, S., Sohel, F., & Boussaid, F. (2017, July). A new representation of skeleton sequences for 3d action recognition. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on* (pp. 4570-4579). IEEE.

Kempen, G. I. J. M., Myers, A. M., & Powell, L. E. (1995). Hierarchical structure in ADL and IADL: analytical assumptions and applications for clinicians and researchers. *Journal of clinical epidemiology*, 48(11), 1299-1305.

Kempen, G. I., & Suurmeijer, T. P. (1990). The development of a hierarchical polychotomous ADL-IADL scale for noninstitutionalized elders. *The Gerontologist*, 30(4), 497-502.

- Kijak, E., Gravier, G., Gros, P., Oisel, L., & Bimbot, F. (2003, July). HMM based structuring of tennis videos using visual and audio cues. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on* (Vol. 3, pp. III-309). IEEE.
- Kim, M., Jeong, C. Y., & Shin, H. C. (2018, October). Activity Recognition using Fully Convolutional Network from Smartphone Accelerometer. In *2018 International Conference on Information and Communication Technology Convergence (ICTC)* (pp. 1482-1484). IEEE.
- Kingma, D. P., & Ba, J. (2014). Adam: a method for stochastic optimization. *CoRR abs/1412.6980* (2014).
- Knickman, J. R., & Snell, E. K. (2002). The 2030 problem: caring for aging baby boomers. *Health services research, 37*(4), 849-884.
- Kolekar, M. H., & Dash, D. P. (2016, November). Hidden markov model based human activity recognition using shape and optical flow based features. In *Region 10 Conference (TENCON), 2016 IEEE* (pp. 393-397). IEEE.
- Krishnan, N. C., & Cook, D. J. (2014). Activity recognition on streaming sensor data. *Pervasive and mobile computing, 10*, 138-154.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).
- Kudo, M., Toyama, J., & Shimbo, M. (1999). Multidimensional curve classification using passing-through regions. *Pattern Recognition Letters, 20*(11-13), 1103-1111.
- Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., & Serre, T. (2011, November). HMDB: a large video database for human motion recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on* (pp. 2556-2563). IEEE.
- Kulkarni, P. (2012). Introduction to reinforcement and systemic machine learning. *Reinforcement and systemic machine learning for decision making, 1-21*.
- Kurakin, A., Zhang, Z., & Liu, Z. (2012, August). A real time system for dynamic hand gesture recognition with a depth sensor. In *EUSIPCO* (Vol. 2, No. 5, p. 6).

Kwolek, B., & Kepski, M. (2014). Human fall detection on embedded platform using depth maps and wireless accelerometer. *Computer methods and programs in biomedicine*, 117(3), 489-501.

Lafferty, J., McCallum, A., & Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Lawton, M. P. (1990). Aging and performance of home tasks. *Human factors*, 32(5), 527-536.

LeCun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10), 1995.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436.

Lee, S. M., Yoon, S. M., & Cho, H. (2017, February). Human activity recognition from accelerometer data using Convolutional Neural Network. In *Big Data and Smart Computing (BigComp), 2017 IEEE International Conference on* (pp. 131-134). IEEE.

Li, K., & Fu, Y. (2012, November). ARMA-HMM: A new approach for early recognition of human activity. In *Pattern Recognition (ICPR), 2012 21st International Conference on* (pp. 1779-1782). IEEE.

Li, L., Zhao, Y., Jiang, D., Zhang, Y., Wang, F., Gonzalez, I., ... & Sahli, H. (2013, September). Hybrid Deep Neural Network--Hidden Markov Model (DNN-HMM) Based Speech Emotion Recognition. In *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on* (pp. 312-317). IEEE.

Liang, S., & Srikant, R. (2016). Why deep neural networks for function approximation?. *arXiv preprint arXiv:1610.04161*.

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In *European conference on computer vision* (pp. 740-755). Springer, Cham.

Lin, W., Sun, M. T., Poovandran, R., & Zhang, Z. (2008, May). Human activity recognition for video surveillance. In *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on* (pp. 2737-2740). IEEE.

Linjordet, T., & Balog, K. (2019, April). Impact of Training Dataset Size on Neural Answer Selection Models. In *European Conference on Information Retrieval* (pp. 828-835). Springer, Cham.

Liu, B., & Ferrari, V. (2017, October). Active learning for human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 4363-4372).

Liu, K., Chen, C., Jafari, R., & Kehtarnavaz, N. (2014, October). Multi-HMM classification for hand gesture recognition using two differing modality sensors. In *Circuits and Systems Conference (DCAS), 2014 IEEE Dallas* (pp. 1-4). IEEE.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431-3440).

Ma, C. Y., Chen, M. H., Kira, Z., & AlRegib, G. (2019). Ts-lstm and temporal-inception: Exploiting spatiotemporal dynamics for activity recognition. *Signal Processing: Image Communication*, 71, 76-87.

Ma, M., Fan, H., & Kitani, K. M. (2016). Going deeper into first-person activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1894-1903).

Martins, P., & Batista, J. (2009, November). Identity and expression recognition on low dimensional manifolds. In *ICIP* (pp. 3341-3344).

Mayer, R. E., & Sims, V. K. (1994). For whom is a picture worth a thousand words? Extensions of a dual-coding theory of multimedia learning. *Journal of educational psychology*, 86(3), 389.

McCain, K. W. (1990). Mapping authors in intellectual space: A technical overview. *Journal of the American society for information science*, 41(6), 433-443.

McCallum, A., & Nigam, K. (1998, July). A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization* (Vol. 752, No. 1, pp. 41-48).

McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (2006). A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI magazine*, 27(4), 12.

Mehr, H. D., Polat, H., & Cetin, A. (2016, April). Resident activity recognition in smart homes by using artificial neural networks. In *Smart Grid Congress and Fair (ICSG), 2016 4th International Istanbul* (pp. 1-5). IEEE.

Meinel, L., Findeisen, M., Hes, M., Apitzsch, A., & Hirtz, G. (2014, January). Automated real-time surveillance for ambient assisted living using an omnidirectional camera. In *Consumer Electronics (ICCE), 2014 IEEE International Conference on* (pp. 396-399). IEEE.

Miao, Y., Gowayyed, M., & Metze, F. (2015, December). EESN: End-to-end speech recognition using deep RNN models and WFST-based decoding. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)* (pp. 167-174). IEEE.

Mindrila, D., & Balentyne, P. (2017). Scatterplots and correlation. Retrieved from.

Mlinac, M. E., & Feng, M. C. (2016). Assessment of activities of daily living, self-care, and independence. *Archives of Clinical Neuropsychology*, 31(6), 506-516.

Mobark, M., Chuprat, S., & Mantoro, T. (2017, November). Improving the accuracy of complex activities recognition using accelerometer-embedded mobile phone classifiers. In *Informatics and Computing (ICIC), 2017 Second International Conference on* (pp. 1-5). IEEE.

Mohamed, R., Perumal, T., Sulaiman, M. N., & Mustapha, N. (2017). Multi Resident Complex Activity Recognition in Smart Home: A Literature Review. *Int. J. Smart Home*, 11(6), 21-32.

Mohamed, R., Perumal, T., Sulaiman, M., Mustapha, N., & Zainudin, M. N. (2018). Multi label classification on multi resident in smart home using classifier chains. *Advanced Science Letters*, 24(2), 1316-1319.

Moore, D. S., & Kirkland, S. (2007). *The basic practice of statistics (Vol. 2)*. New York: WH Freeman

Moy, M. L., Mentzer, S. J., & Reilly, J. J. (2003). Ambulatory monitoring of cumulative free-living activity. *IEEE engineering in medicine and biology magazine*, 22(3), 89-95.

Mozer, M. C. (1998, March). The neural network house: An environment that adapts to its inhabitants. In Proc. AAAI Spring Symp. Intelligent Environments (Vol. 58).

Munther, A., Othman, R. R., Alsaadi, A. S., & Anbar, M. (2016). A performance study of hidden Markov model and random forest in internet traffic classification. In Information Science and Applications (ICISA) 2016 (pp. 319-329). Springer, Singapore.

Nefian, A. V., & Hayes, M. H. (1998, October). Face detection and recognition using hidden Markov models. In Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on (Vol. 1, pp. 141-145). IEEE.

Nguyen, A., Yosinski, J., & Clune, J. (2015). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 427-436).

Niu, F., & Abdel-Mottaleb, M. (2005, July). HMM-based segmentation and recognition of human activities from video sequences. In 2005 IEEE International Conference on Multimedia and Expo (pp. 804-807). IEEE.

Norris, C., & Moran, J. (2016). Surveillance, closed circuit television and social control. Routledge.

Olah, C. (2015). Understanding lstm networks.

Oliver, N. M., Rosario, B., & Pentland, A. P. (2000). A Bayesian computer vision system for modeling human interactions. IEEE transactions on pattern analysis and machine intelligence, 22(8), 831-843.

Ordóñez, F. J., de Toledo, P., & Sanchis, A. (2013). Activity recognition using hybrid generative/discriminative models on home environments using binary sensors. Sensors, 13(5), 5460-5477.

Ordóñez, F., & Roggen, D. (2016). Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. Sensors, 16(1), 115.

Palm, C. S. (1997). U.S. Patent No. 5,699,444. Washington, DC: U.S. Patent and Trademark Office.

- Park, S., & Jayaraman, S. (2003). Enhancing the quality of life through wearable technology. *IEEE Engineering in medicine and biology magazine*, 22(3), 41-48.
- Park, S., Ji, M., & Chun, J. (2018). 2D Human Pose Estimation based on Object Detection using RGB-D information. *KSII Transactions on Internet & Information Systems*, 12(2).
- Patel, S., & Chauhan, Y. (2014). Heart attack detection and medical attention using motion sensing device-kinect. *International Journal of Scientific and Research Publications*, 4(1).
- Patzold, M. (2019). 5G Is Coming Around the Corner [Mobile Radio]. *IEEE Vehicular Technology Magazine*, 14(1), 4–10. <https://doi.org/10.1109/MVT.2018.2884042>
- Paulus, P. B. (1983). Group influence on individual task performance. In *Basic group processes* (pp. 97-120). Springer, New York, NY.
- Pauly, L., Hogg, D., Fuentes, R., & Peel, H. (2017, July). Deeper networks for pavement crack detection. In *Proceedings of the 34th ISARC* (pp. 479-485). IAARC.
- Perumal, T., Chui, Y. L., Ahmadon, M. A. B., & Yamaguchi, S. (2017, October). IoT based activity recognition among smart home residents. In *Consumer Electronics (GCCE), 2017 IEEE 6th Global Conference on* (pp. 1-2). IEEE.
- Pinquier, J., Karaman, S., Letoupin, L., Guyot, P., Mégret, R., Benois-Pineau, J., ... & Dartigues, J. F. (2012, November). Strategies for multiple feature fusion with hierarchical hmm: application to activity recognition from wearable audiovisual sensors. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)* (pp. 3192-3195). IEEE.
- Poorani, M., Vaidehi, V., & Varalakshmi, P. (2017, January). Performance analysis of triaxial accelerometer for activity recognition. In *Advanced Computing (ICoAC), 2016 Eighth International Conference on* (pp. 170-175). IEEE.
- Priddy, K. L., & Keller, P. E. (2005). *Artificial neural networks: an introduction* (Vol. 68). SPIE press.
- Qian, K., Ma, X., Dai, X., & Hu, C. (2008, June). A multi-camera approach to tracking and localization of people with coexisting robots. In *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on* (pp. 5162-5167). IEEE.

Quattoni, A., Wang, S., Morency, L. P., Collins, M., & Darrell, T. (2007). Hidden conditional random fields. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (10), 1848-1852.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), 257-286.

Raeiszadeh, M., & Tahayori, H. (2018, February). A novel method for detecting and predicting resident's behavior in smart home. In *Fuzzy and Intelligent Systems (CFIS), 2018 6th Iranian Joint Congress on* (pp. 71-74). IEEE.

Ramadijanti, N., Fahrul, H. F., & Pangestu, D. M. (2016, November). Basic dance pose applications using kinect technology. In *Knowledge Creation and Intelligent Computing (KCIC), International Conference on* (pp. 194-200). IEEE.

Ramakrishna, V., Munoz, D., Hebert, M., Bagnell, J. A., & Sheikh, Y. (2014, September). Pose machines: Articulated pose estimation via inference machines. In *European Conference on Computer Vision* (pp. 33-47). Springer, Cham.

Rashidi, P., & Mihailidis, A. (2013). A survey on ambient-assisted living tools for older adults. *IEEE journal of biomedical and health informatics*, 17(3), 579-590.

Ravi, D., Wong, C., Deligianni, F., Berthelot, M., Andreu-Perez, J., Lo, B., & Yang, G. Z. (2017). Deep learning for health informatics. *IEEE journal of biomedical and health informatics*, 21(1), 4-21.

Reddy, K. K., & Shah, M. (2013). Recognizing 50 human action categories of web videos. *Machine Vision and Applications*, 24(5), 971-981.

Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7263-7271).

Rendle, S. (2010, December). Factorization machines. In *2010 IEEE International Conference on Data Mining* (pp. 995-1000). IEEE.

Riis, S. K., & Krogh, A. (1997, April). Hidden neural networks: A framework for HMM/NN hybrids. In *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on* (Vol. 4, pp. 3233-3236). IEEE.

Rimminen, H., Lindström, J., Linnavuo, M., & Sepponen, R. (2010). Detection of falls among the elderly by a floor sensor using the electric near field. *IEEE Transactions on Information Technology in Biomedicine*, 14(6), 1475-1476.

Rohrbach, M., Amin, S., Andriluka, M., & Schiele, B. (2012, June). A database for fine grained activity detection of cooking activities. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on* (pp. 1194-1201). IEEE.

Roser, M., Ortiz-Ospina, E., & Ritchie, H. (2013). Life expectancy. *Our World in Data*.

Rössl, C., Kobbelt, L., & Seidel, H.-P. (2000). Extraction of feature lines on triangulated surfaces using morphological operators. Retrieved from www.aai.org

Ruder, S. (2016). An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.

Rynkiewicz, J. (1999, April). Hybrid HMM/MLP models for times series prediction. In *ESANN* (pp. 455-462).

Safavian, S. R., & Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, 21(3), 660-674.

Salzman, B. (2010). Gait and balance disorders in older adults. *Am Fam Physician*, 82(1), 61-68.

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3), 210-229.

Sapp, B., & Taskar, B. (2013). Modec: Multimodal decomposable models for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3674-3681).

Sauseng, P., & Klimesch, W. (2008). What does phase information of oscillatory brain activity tell us about cognitive processes?. *Neuroscience & Biobehavioral Reviews*, 32(5), 1001-1013.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61, 85-117.

Schröder, J., Anemüller, J., & Goetze, S. (2016). Performance comparison of GMM, HMM and DNN based approaches for acoustic event detection within task 3 of the DCASE 2016 challenge. In Proc. Workshop Detect. Classification Acoust. Scenes Events (pp. 80-84).

Scuffham, P., Chaplin, S., & Legood, R. (2003). Incidence and costs of unintentional falls in older people in the United Kingdom. *Journal of Epidemiology & Community Health*, 57(9), 740-744.

Sedgwick, P. (2012). Pearson's correlation coefficient. *Bmj*, 345, e4483.

Sermanet, P., Chintala, S., & LeCun, Y. (2012, November). Convolutional neural networks applied to house numbers digit classification. In *Pattern Recognition (ICPR), 2012 21st International Conference on* (pp. 3288-3291). IEEE.

Shen, W., Deng, K., Bai, X., Leyvand, T., Guo, B., & Tu, Z. (2014). Exemplar-based human action pose correction. *IEEE transactions on cybernetics*, 44(7), 1053-1066.

Shimada, A., Kondo, K., Deguchi, D., Morin, G., & Stern, H. (2013). Kitchen scene context based gesture recognition: A contest in ICPR2012. In *Advances in depth image analysis and applications* (pp. 168-185). Springer, Berlin, Heidelberg.

Shu, T., Todorovic, S., & Zhu, S. C. (2017, July). CERN: confidence-energy recurrent network for group activity recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (Vol. 2)*.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Soekhoe, D., Van Der Putten, P., & Plaat, A. (2016, October). On the impact of data set size in transfer learning using deep neural networks. In *International Symposium on Intelligent Data Analysis* (pp. 50-60). Springer, Cham.

Solaimanpour, S., & Doshi, P. (2017, May). A layered HMM for predicting motion of a leader in multi-robot settings. In *2017 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 788-793). IEEE.

Soomro, K., Zamir, A. R., & Shah, M. (2012). UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Stadermann, J., & Rigoll, G. (2004). A hybrid SVM/HMM acoustic modeling approach to automatic speech recognition. In Proc. Int. Conf. on Spoken Language Processing ICSLP# 2004, Jeju Island, South Korea.

Stommel, M., Beetz, M., & Xu, W. (2015). Model-Free detection, encoding, retrieval, and visualization of human poses from kinect data. *IEEE/ASME Transactions on Mechatronics*, 20(2), 865-875.

Sung, J., Ponce, C., Selman, B., & Saxena, A. (2011, August). Human activity detection from RGBD images. In *Workshops at the twenty-fifth AAAI conference on artificial intelligence*.

Sung, J., Ponce, C., Selman, B., & Saxena, A. (2012, May). Unstructured human activity detection from rgbd images. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on* (pp. 842-849). IEEE.

Suykens, J. A., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural processing letters*, 9(3), 293-300.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

Szegedy, C., Toshev, A., & Erhan, D. (2013). Deep neural networks for object detection. In *Advances in neural information processing systems* (pp. 2553-2561).

Takač, B., Català, A., Martín, D. R., Van Der Aa, N., Chen, W., & Rauterberg, M. (2013). Position and orientation tracking in a ubiquitous monitoring system for Parkinson disease patients with freezing of gait symptom. *JMIR mHealth and uHealth*, 1(2).

Tao, L., Burghardt, T., Hannuna, S., Camplani, M., Paiement, A., Damen, D., ... & Craddock, I. (2015, October). A comparative home activity monitoring study using visual and inertial sensors. In *E-health Networking, Application & Services (HealthCom), 2015 17th International Conference on* (pp. 644-647). IEEE.

Tenorth, M., Bandouch, J., & Beetz, M. (2009, September). The TUM kitchen data set of everyday manipulation activities for motion tracking and action recognition. In *Computer*

Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on (pp. 1089-1096). IEEE.

Thys, S., Van Ranst, W., & Goedemé, T. (2019). Fooling automated surveillance cameras: adversarial patches to attack person detection. arXiv preprint arXiv:1904.08653.

Tian, Y., Zitnick, C. L., & Narasimhan, S. G. (2012, October). Exploring the spatial hierarchy of mixture models for human pose estimation. In European Conference on Computer Vision (pp. 256-269). Springer, Berlin, Heidelberg.

Tome, D., Russell, C., & Agapito, L. (2017). Lifting from the deep: Convolutional 3d pose estimation from a single image. CVPR 2017 Proceedings, 2500-2509.

Töreyn, B. U., Dedeoğlu, Y., & Çetin, A. E. (2005, October). HMM based falling person detection using both audio and video. In International Workshop on Human-Computer Interaction (pp. 211-220). Springer, Berlin, Heidelberg.

Touretzky, D. S., Mozer, M. C., & Hasselmo, M. E. (Eds.). (1996). Advances in Neural Information Processing Systems 8: Proceedings of the 1995 Conference (Vol. 8). Mit Press.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE international conference on computer vision (pp. 4489-4497).

Triandis, H. C. (1994). Culture and social behavior.

Tsai, A. C., Ou, Y. Y., Sun, C. A., & Wang, J. F. (2017, December). VQ-HMM classifier for human activity recognition based on R-GBD sensor. In Orange Technologies (ICOT), 2017 International Conference on (pp. 201-204). IEEE.

Uddin, M. Z., Thang, N. D., & Kim, T. S. (2010, September). Human Activity Recognition via 3-D joint angle features and Hidden Markov models. In Image Processing (ICIP), 2010 17th IEEE International Conference on (pp. 713-716). IEEE.

Uddin, M. Z., Torresen, J., & Jabid, T. (2016, October). Human activity recognition using depth body part histograms and hidden markov models. In Innovations in Science, Engineering and Technology (ICISSET), International Conference on (pp. 1-4). IEEE.

Vajda, T., & Zoltán, Á. (2011, August). Pictorial structure based people detection and pose estimation in videos. In *Intelligent Computer Communication and Processing (ICCP), 2011 IEEE International Conference on* (pp. 315-318). IEEE.

Van Kasteren, T., Noulas, A., Englebienne, G., & Kröse, B. (2008, September). Accurate activity recognition in a home setting. In *Proceedings of the 10th international conference on Ubiquitous computing* (pp. 1-9). ACM.

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *CVPR (1)*, 1, 511-518.

Vydana, H. K., Pulugandla, B., Shrivastava, M., & Vuppala, A. K. (2017, December). DNN-HMM Acoustic Modeling for Large Vocabulary Telugu Speech Recognition. In *Mining Intelligence and Knowledge Exploration: 5th International Conference, MIKE 2017, Hyderabad, India, December 13–15, 2017, Proceedings* (Vol. 10682, p. 189). Springer.

Waltner, G., Mauthner, T., & Bischof, H. (2014). Improved Sport Activity Recognition using Spatio-temporal Context. In *Proc. DVS-Conference on Computer Science in Sport (DVS/GSSS)*.

Wang, J., Zhang, X., Gao, Q., Yue, H., & Wang, H. (2017). Device-free wireless localization and activity recognition: A deep learning approach. *IEEE Transactions on Vehicular Technology*, 66(7), 6258-6267.

Wang, L., Xiong, Y., Wang, Z., & Qiao, Y. (2015). Towards good practices for very deep two-stream convnets. *arXiv preprint arXiv:1507.02159*.

Wang, S. B., Quattoni, A., Morency, L. P., Demirdjian, D., & Darrell, T. (2006). Hidden conditional random fields for gesture recognition. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)* (Vol. 2, pp. 1521-1527). IEEE.

Wang, X., Takaki, S., & Yamagishi, J. (2016, September). A comparative study of the performance of HMM, DNN, and RNN based speech synthesis systems trained on very large speaker-dependent corpora. In *9th ISCA Speech Synthesis Workshop* (Vol. 9, pp. 125-128).

Wang, Y., Huang, K., & Tan, T. (2007, September). Abnormal activity recognition in office based on R transform. In *Image Processing, 2007. IICIP 2007. IEEE International Conference on* (Vol. 1, pp. 1-341). IEEE.

Ward, J. A., Lukowicz, P., Troster, G., & Starner, T. E. (2006). Activity recognition of assembly tasks using body-worn microphones and accelerometers. *IEEE transactions on pattern analysis and machine intelligence*, 28(10), 1553-1567.

Waterhouse, E. (2003). New horizons in ambulatory electroencephalography. *IEEE Engineering in Medicine and Biology Magazine*, 22(3), 74-80.

Wei, S. E., Ramakrishna, V., Kanade, T., & Sheikh, Y. (2016). Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4724-4732).

White, H. D., & McCain, K. W. (1998). Visualizing a discipline: An author co-citation analysis of information science, 1972–1995. *Journal of the American society for information science*, 49(4), 327-355.

Williams, A., Xie, D., Ou, S., Grupen, R., Hanson, A., & Riseman, E. (2006). Distributed smart cameras for aging in place. MASSACHUSETTS UNIV AMHERST DEPT OF COMPUTER SCIENCE.

Witten, I. H., & Frank, E. (2002). Data mining: practical machine learning tools and techniques with Java implementations. *Acm Sigmod Record*, 31(1), 76-77.

Wu, H., Pan, W., Xiong, X., & Xu, S. (2014, July). Human activity recognition based on the combined svm&hmm. In *Information and Automation (ICIA), 2014 IEEE International Conference on* (pp. 219-224). IEEE.

Wu, Z., Jiang, Y. G., Wang, X., Ye, H., Xue, X., & Wang, J. (2015). Fusing multi-stream deep networks for video classification. *arXiv preprint arXiv:1509.06086*.

Yamato, J., Ohya, J., & Ishii, K. (1992, June). Recognizing human action in time-sequential images using hidden markov model. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on* (pp. 379-385). IEEE.

Yang, J., Xu, Y., & Chen, C. S. (1997). Human action learning via hidden Markov model. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 27(1), 34-44.

Yang, Y., & Ramanan, D. (2013). Articulated human detection with flexible mixtures of parts. *IEEE transactions on pattern analysis and machine intelligence*, 35(12), 2878-2890.

Yin, X., Wang, B., Li, W., Liu, Y., & Zhang, M. (2015). Background subtraction for moving cameras based on trajectory-controlled segmentation and label inference. *KSII Transactions on Internet and Information Systems*, 9(10), 4092–4107. <https://doi.org/10.3837/tiis.2015.10.018>

Yogesh, K. M. (2017, September). Instance based human physical activity (hpa) recognition using shimmer2 wearable sensor data sets. In *Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on* (pp. 995-999). IEEE.

Youness, C., & Abdelhak, M. (2016, March). Machine learning for real time poses classification using Kinect skeleton data. In *2016 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV)* (pp. 307-311). IEEE.

Yu, H. Y., Chen, J. J., & Kuo, C. H. (2014, June). The human-environment interface design with a vision assistance module or a smart wheelchair. In *Advanced Robotics and Intelligent Systems (ARIS), 2014 International Conference on* (pp. 91-96). IEEE.

Yun, K., Honorio, J., Chattopadhyay, D., Berg, T. L., & Samaras, D. (2012, June). Two-person interaction detection using body-pose features and multiple instance learning. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on* (pp. 28-35). IEEE.

Zhang, L., Wu, X., & Luo, D. (2015, July). Human activity recognition with HMM-DNN model. In *Cognitive Informatics & Cognitive Computing (ICCI* CC), 2015 IEEE 14th International Conference on* (pp. 192-197). IEEE.

Zhou, Z., Stone, E. E., Skubic, M., Keller, J., & He, Z. (2011, August). Nighttime in-home action monitoring for eldercare. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE* (pp. 5299-5302). IEEE.

Zhu, W., Lan, C., Xing, J., Zeng, W., Li, Y., Shen, L., & Xie, X. (2016, March). Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Appendix A

FORM UPR16

Research Ethics Review Checklist

Please include this completed form as an appendix to your thesis (see the Research Degrees Operational Handbook for more information)



Postgraduate Research Student (PGRS) Information		Student ID:	444947
PGRS Name:	Mohamad Zuhair Saeed Al-Wattar		
Department:	SoC	First Supervisor:	Dr Rinat Khusainov
Start Date: (or progression date for Prof Doc students)	01/02/2015		
Study Mode and Route:	Part-time <input type="checkbox"/>	MPhil <input type="checkbox"/>	MD <input type="checkbox"/>
	Full-time <input checked="" type="checkbox"/>	PhD <input checked="" type="checkbox"/>	Professional Doctorate <input type="checkbox"/>

Title of Thesis:	Context Fusion for Recognising Activities of Daily Living from Indoor Video Data
Thesis Word Count: (excluding ancillary data)	44370

If you are unsure about any of the following, please contact the local representative on your Faculty Ethics Committee for advice. Please note that it is your responsibility to follow the University's Ethics Policy and any relevant University, academic or professional guidelines in the conduct of your study

Although the Ethics Committee may have given your study a favourable opinion, the final responsibility for the ethical conduct of this work lies with the researcher(s).

UKRIO Finished Research Checklist:

(If you would like to know more about the checklist, please see your Faculty or Departmental Ethics Committee rep or see the online version of the full checklist at: <http://www.ukrio.org/what-we-do/code-of-practice-for-research/>)


a) Have all of your research and findings been reported accurately, honestly and within a reasonable time frame?	YES <input type="checkbox"/>	NO <input checked="" type="checkbox"/>
b) Have all contributions to knowledge been acknowledged?	YES <input type="checkbox"/>	NO <input checked="" type="checkbox"/>
c) Have you complied with all agreements relating to intellectual property, publication and authorship?	YES <input type="checkbox"/>	NO <input checked="" type="checkbox"/>
d) Has your research data been retained in a secure and accessible form and will it remain so for the required duration?	YES <input type="checkbox"/>	NO <input checked="" type="checkbox"/>
e) Does your research comply with all legal, ethical, and contractual requirements?	YES <input type="checkbox"/>	NO <input checked="" type="checkbox"/>

Candidate Statement:

I have considered the ethical dimensions of the above named research project, and have successfully obtained the necessary ethical approval(s)

Ethical review number(s) from Faculty Ethics Committee (or from NRES/SCREC): ETHIC-1019-67

If you have *not* submitted your work for ethical review, and/or you have answered 'No' to one or more of questions a) to e), please explain below why this is so:

Signed (PGRS):  **Date:** 16/04/2020