Smart Home Nursing

Session: 2018 – 2022

Submitted by:

Anamta Khan	2018-CS-10
Sidra Khan	2018-CS-16
Maria Azrar	2018-CS-50
Alaina Faisal	2018-CS-55

Supervised by:

Mr. Samyan Qayyum Wahla

Department of Computer Science

University of Engineering and Technology

Lahore Pakistan

Smart Home Nursing

Submitted to the Faculty of the Computer Science Department of the University of Engineering and Technology Lahore in partial fulfillment of the requirements for the Degree of

Bachelor of Science

in

Computer Science.

Internal Examiner	**External Examiner**
Signature:	Signature:
Mr. Samyan Qayyum Wahla	Name:
Lecturer	Designation:
Chairman	**Dean**
Signature:	Signature:
Prof. Dr. Muhammad Shoaib	Prof. Dr. Muhammad Kamran

Department of Computer Science

University of Engineering and Technology

Lahore Pakistan

Declaration

We declare that the work contained in this thesis is our own, except where explicitly stated otherwise. In addition this work has not been submitted to obtain another degree or professional qualification.

Signed: _____ Signed: _____

Date: _____ Date: _____

Signed: _____ Signed: _____

Date: _____ Date: _____

Acknowledgments

First and foremost, we owe a debt of thanks to the Almighty Allah for providing us with the strength to complete this task. It brings us great pleasure to convey our sincere thanks and respect to Mr. Samyan Qayyum Wahla, our supervisor, for instilling confidence and excitement in us and inspiring us in our work through his encouragement and advice. Our warmest gratitude to him for his invaluable advice and recommendations. As his pupils, we present this dissertation work with great pride and joy.

Next, we'd want to express our gratitude to our family for providing the environment and motivation to work on this project. Finally, but certainly not least, we want to express our gratitude to our parents for their unconditional love, devotion, kind cooperation, and support, without which we would not have been able to complete this project.

Dedication

Dedicated to Allah, the Almighty in gratitude for His favours and kindness. We'd like to dedicate this thesis and Final Year Project to our loving family and friends who have always been there for us.

Contents

Acknowledgments iii

Dedication iv

List of Figures ix

List of Tables xi

Abbreviations xii

Abstract xiii

1 Introduction 2
- 1.1 Overview of Project . . . 2
- 1.2 Background . . . 3
- 1.3 Objectives . . . 4
  - 1.3.1 Industry Objectives . . . 4
  - 1.3.2 Research Objectives . . . 4
  - 1.3.3 Academic Objectives . . . 5
- 1.4 Scope and Applications . . . 5
- 1.5 Target Audience . . . 7
  - 1.5.1 Elderly People . . . 7
  - 1.5.2 Patients . . . 7
  - 1.5.3 Hospitals . . . 7
  - 1.5.4 Researchers . . . 7
  - 1.5.5 Businessmen . . . 7
  - 1.5.6 Health Care Professionals . . . 8
  - 1.5.7 Psychologists . . . 8
- 1.6 System Requirements . . . 8
  - 1.6.1 Hardware Requirements . . . 8
  - 1.6.2 Software Requirements . . . 8
- 1.7 Feasibility Analysis . . . 9
  - 1.7.1 Technical Feasibility . . . 9
  - 1.7.2 Operational Feasibility . . . 9
  - 1.7.3 Economical Feasibility . . . 9
- 1.8 Challenges and Limitations . . . 9

1.8.1 View-Point Variation . . . 10
1.8.2 Occlusion . . . 10
1.8.3 Illumination . . . 10
1.8.4 Background Clutter . . . 11
1.8.5 Data collection . . . 11
1.8.6 Camera Synchronization . . . 11
1.8.7 Limitations . . . 12

2 Literature Survey **13**
2.1 Literature Review . . . 13
2.2 Motivation . . . 19
2.3 Problem Statement . . . 19
2.4 Our Contribution . . . 19

3 Dataset **22**
3.1 Existing Datasets . . . 22
3.1.1 Kinetics 400 . . . 22
3.1.2 ActivityNet . . . 23
3.1.3 Kinetics 600 . . . 23
3.1.4 Kinetics 700 . . . 24
3.1.5 AVA-Kinetics . . . 24
3.1.6 HACS (MIT) . . . 25
3.1.7 MMACT . . . 25
3.1.8 Home Action Genome . . . 26
3.1.9 SoccerNet . . . 26
3.1.10 VidSitu . . . 26
3.1.11 Lifelog Dataset . . . 27
3.1.12 IKEA ASM . . . 27
3.1.13 WISDM . . . 27
3.1.14 Opportunity Dataset . . . 28
3.1.15 UCI-HAR Dataset . . . 28
3.1.16 USC-HAD Dataset . . . 29
3.1.17 Skoda Dataset . . . 30
3.1.18 PAMAP2 Dataset . . . 31
3.1.19 DAPHNET Dataset . . . 31
3.1.20 Rose NTU Dataset . . . 32
3.1.21 Toyota Smart Dataset . . . 32
3.1.22 UR Fall Dataset . . . 33
3.2 Local Dataset . . . 37
3.2.1 Dataset Details . . . 37
3.2.2 Data Annotation . . . 37
3.2.3 Naming Convention . . . 38

4 Proposed Methodology **39**
4.1 Activity Detection using Optical Flow . . . 40

4.1.1 Dataset . . . 40
4.1.2 Dataset Division . . . 40
4.1.3 Optical Flow . . . 40
4.1.4 Multi-Layer Perceptron (MLP) Model . . . 42
4.1.5 Results . . . 44
4.2 Gait Based Activity Recognition . . . 45
4.2.1 Background Subtraction . . . 45
4.2.2 Skeletonization . . . 46
4.2.3 Dividing Human Skeleton . . . 46
4.2.4 Hough Transform and Feature Extraction . . . 47
4.2.5 Classification . . . 48
4.2.6 Dataset . . . 50
4.2.7 Results . . . 51
4.3 Activity Recognition using RNN . . . 53
4.3.1 Dataset . . . 53
4.3.2 Frames Generation . . . 55
4.3.3 RNN Architecture . . . 55
4.3.4 Results . . . 58
4.4 Activity Recognition using LSTM . . . 60
4.4.1 Dataset . . . 60
4.4.2 Dropout for LSTM . . . 62
4.4.3 LSTM Architecture . . . 62
4.4.4 Results . . . 64
4.5 Activity Recognition using LSTM-CNN . . . 66
4.5.1 Dataset . . . 66
4.5.2 Extracting Frames . . . 68
4.5.3 Extracting Features . . . 68
4.5.4 LSTM-CNN Architecture . . . 68
4.5.5 LSTM Layers . . . 69
4.5.6 Convolutional and Pooling Layer . . . 70
4.5.7 Training Model . . . 70
4.5.8 Results . . . 71
4.6 Predictive Modeling . . . 74
4.6.1 Data preparation . . . 75
4.6.2 Stacked LSTM . . . 75
4.6.3 Results . . . 75
4.7 Anomaly detection using an Autoencoder . . . 76
4.7.1 Preparing data . . . 76
4.7.2 Model . . . 76
4.7.3 Results . . . 76
4.8 Trend analysis . . . 77
4.8.1 Results . . . 79

5 Conclusion **80**

5.1 Conclusion 80
5.1.1 Future Work 80

References **82**

List of Figures

1.1 View-Point Variation 10
1.2 Occlusion 10
1.3 Illumination 11

3.1 Videos in Personal Dataset 37

4.1 General Flow Diagram 39
4.2 Folder Structure for Dataset 40
4.3 Results Obtained from Optical Flow 41
4.4 Frame Dimension 42
4.5 Typical Architecture of the MLP Model 43
4.6 Rectified Linear Unit (ReLu) Activation Function 44
4.7 methodology of Gait Based Activity Detection 45
4.8 Applying Median Filter 45
4.9 Background Subtraction 46
4.10 Skeleton using Medial Axis Morphology 46
4.11 Skeleton Divison 46
4.12 Lines Detected using Hough Transform 47
4.13 Gait Features Extraction 48
4.14 Example of K-Nearest Neighbour 48
4.15 Example of Decision tree 49
4.16 KTH Dataset 50
4.17 Folder Structure for Dataset 50
4.18 Confusion Matrix of KNN 53
4.19 Training dataset structure 54
4.20 Testing dataset structure 54
4.21 Frames of class Laydown 55
4.22 RNN Architecture Diagram 56
4.23 Confusion Matrix 59
4.24 Training Accuracy 60
4.25 Training Loss 60
4.26 Training dataset structure 61
4.27 Testing dataset structure 62
4.28 LSTM Neuron 63
4.29 LSTM Architecture 63
4.30 Confusion Matrix 65

4.31 Training accuracy 66
4.32 Training Loss 66
4.33 Training dataset structure 67
4.34 Testing dataset structure 67
4.35 Inception V3 Architecture 68
4.36 LSTM-CNN Architecture Diagram 69
4.37 Confusion Matrix 72
4.38 Training Accuracy 73
4.39 Training Loss 73
4.40 ROC Curve 74
4.41 Labels for ROC 74
4.42 Result for predictive model 76
4.43 Training accuracy and loss 77
4.44 Anomaly Detection 77
4.45 Sleeping hours over different days 78
4.46 No. of people awake at different times 78
4.47 Trend for sleep 79

List of Tables

2.1 Comparison of state-of-art work 20

3.1 Comparison of Existing Datasets 34

4.1 Results Obtained from the MLP Model 44
4.2 Training and Testing samples of each class 51
4.3 Comparison of different classification algorithms 52
4.4 Results Of RNN 58
4.5 Results of LSTM 65
4.6 Results of LSTM-CNN 72

Abbreviations

NLU	**N**atural **L**anguage **U**nderstanding
ML	**M**achine **L**earning
LSTM	**L**ong **S**hort **T**erm **M**emory
BN	**B**ayesian **N**etworks
DT	**D**ecision **T**rees
SVM	**S**upport **V**ector **M**achine
RNN	**R**ecurrent **N**eural **N**etwork
CNN	**C**onvolutional **N**eural **N**etwork

Abstract

Smart Home Nursing helps the elderly infer their daily life activities to ensure their safety and successful ageing. An activity recognition system that classifies a collection of common daily activities and common emergency activities is provided in this paper. It keeps a record of the activities to detect anomalies and provide behavior analysis. It includes commercial, reliable, and well-known video capture gear and is designed to be comfortable and non-intrusive. A deep learning hybrid model for activity recognition with calibration is suggested and compared to other deep convolutional neural network and recurrent neural network models. The proposed approach has a high identification accuracy and allows new users to adapt quickly. The methodology is evaluated using data from 50 individuals and two previously published datasets, and it produces superior results.

Chapter 1

Introduction

1.1 Overview of Project

The main purpose is to use technology for behavioral monitoring to provide elderly assistance by regulating the daily tasks and detecting any abnormal activities or behavior. Routine behavior observance can be used to provide help in performing daily tasks and identify different aspects of their lives to better their standard of living. The activity patterns help in detection of abnormal activities to raise alarm[1].

In the project, daily life activities (DLA) monitoring is one of the essential components. Cameras are to be used to continuously monitor and record the activities of its dwellers. Activities are recorded as symbolic representations to provide different services in consideration of the user's lifestyle and routine behavior. Through machine learning algorithms, falls can be detected in real-time and reported for medical assistance. Similarly, other emergency situations can be detected which require immediate actions and can be detected visually[2]. Activity patterns can also, be learned over time to detect any anomalous behavior such as a change in sleeping patterns, eating patterns, and dementia symptoms.

Early-onset of dementia can be detected by finding symptoms in the daily routine. Similarly, changes in eating patterns can be used to generate a report for possible diseases and to inform the attendant about the anomalous behavior[3]. Assistance in this form can help to provide a capacity for living at home for aged people. It can also provide care for patients with chronic diseases[4].

1.2 Background

The number of elderly people living independently is increasing globally. In the next four decades, the number of people aged 65 and up is expected to rise to 22% of the overall population[5]. This increase is partially due to advancements in medicine, healthcare, and an awareness of a healthy and nutritious lifestyle. According to World Bank, around 15% of the world's population has a disability, which can range from physical limitations to cognitive impairment[6]. People with health issues and disabilities require assistance for daily tasks. Therefore, due to increasing life expectancy and health diseases at older age, there has been an increase in the older population. Due to multiple reasons, the older age group has been moved to live independently or to live in nursing homes[7]. These nursing homes provide a multitude of facilities to improve the elderly's quality of life. Elderly people living alone or in institutions require assistance to perform their daily tasks as well as an immediate action in case of an emergency. It is difficult to monitor their activities all the time and provide the required assistance, especially in the former case[8].

Significant study has recently been conducted in the fields of health risk detection and ageing behaviour in the elderly population. The main focus has been on remote health monitoring and studying behavioral and activity patterns in the elderly[9]. Behavioural patterns can then be used to detect anomalous behaviour and assist if required. Several approaches have been used in the past including wearable devices using IoT, sensors, and depth cameras to collect data. Sensors and IoT provide a way to process patterns without invading the privacy of the people. However, there has been concern regarding usability and usefulness of such sensors[10].

Various health problems and factors are engaged in health monitoring, including patient bodily functions, anxiety, behaviour, social and environmental affects [11], all of which have an impact on each individual's activity pattern. In the present time, huge amount of data has become available which can be used to provide health risk detection. The behavioural pattern can be utilised to improve the automation and detection of elderly's activities in nursing homes using machine learning (ML) approaches. The results can be used to avoid health risks by detection of falls, monitoring of eating and sleeping behaviour, consumption of pills, frequency of bathroom visits, dementia symptoms. Timely intervention can help to save lives and provide a secure smart environment for people[12].

The major purpose is to enable elderly people to live independently at home,

improve their quality of life, and avoid expensive nursing care. Results can be used to improve the quality of life for older individuals as well as providing a long-term cost-effective solution through integration and careful implementation.

1.3 Objectives

1.3.1 Industry Objectives

The following are the industry objectives that will be achieved:

- To improve the functional and practical implementation of existing state of art machine learning methods.
- To provide a better way of an assisted living facility for elderly people living on their own.
- To reduce costs of living independently and provide nursing facilities by installing a mechanism for remote monitoring of the elderly.
- To increase the positive response towards acceptance of smart home nursing by providing easy-to-use software.
- To provide an infrastructure for practical implementation of Smart Home Nursing.

1.3.2 Research Objectives

The following are the research objectives that will be achieved:

- To provide an understanding of algorithms and techniques used in providing assisted living facilities.
- To Identify knowledge gaps in current research and a list of research-able questions with potential.
- To compare the functional state of those who have smart technologies installed in their houses to those who have not made any changes to their homes.
- To identify the area of emphasis for assisted living.
- To provide literature review and comparison by meta-analysis in the emerging area of computer vision to provide substantial research information.

1.3.3 Academic Objectives

The following are the academic objectives that will be achieved:

- To get an understanding of machine learning and computer vision algorithms.
- To understand the difficult technicalities of image processing in real-time.
- To do a cognitive analysis of aging
- To gain an understanding of health risks and how they can be detected

1.4 Scope and Applications

The project will use a Robot WiFi IP Camera with Night Vision placed strategically near the ceiling of the senior's room with a 45◦ downward viewing angle. The cameras use a 1/4 color CMOS sensor with a frame rate of 25FPS. It also provides night vision through 12 infrared LEDs. Other specifications are:

- Resolution: 1280*720(720P), 640*360, 320*180, 160*90 (selectable)
- Lens: 3.6mm

10 types of activities are examined:

- Drinking
- Eating
- Enter
- Lying
- Leave
- Sit down
- Stand up
- Take pills
- Use telephone
- Walk

From these activities, the following events are recognized:

- Eating
- Falls
- Sleeping

We use these events to analyze **sleeping patterns, fall patterns** and **eating patterns**.

Gait analysis is to be used to provide information about mobility level and early neurodegenerative disease risk[13]. Finally, results are to be integrated to build daily activity timelines of seniors and compute relevant analysis.

The following set of modules concretely implements the high-level modules:

- Temporal navigation from coarse to fine (time range and timeline view)
- Description of the subject's behavior at a high level (snapshot)
- Automatic partitioning and description of events (storyboard view)
- Global statistics for the time period chosen (global view)
- Filters and notifications that users can customize

The system can only notify the concerned people about a situation but not intervene in any other way. Communication is also limited to application. It does not take into account vital signs via smart devices such as heart rate or temperature but only visual input in the form of images and videos.

It does not include robotic agents for physical assistance or portable devices. The project is also applicable in other scenarios where assistance in living independently is required by further extension of findings.

The results and methodologies can also be applied in hospital environments to monitor patients in their rooms. Doctors can be informed in case of an emergency after surgery. Nursing staff can also be informed if a patient falls or requires assistance but is not able to call anyone at the moment. Behavioral pattern learning can be applied in medicines to treat several diseases such as those due to sleep. It can also contribute to psychological society in terms of understanding aging behavior and cognitive analysis of sleep patterns. By extension, the findings can be applied to other medical conditions and assist by interacting verbally as well as physically with the person.

1.5 Target Audience

This is a broad field project which may interest people from many fields. The range is wide since it includes research in the computer science field and specifically artificial intelligence, then it also overlaps with home nursing and therefore medical field, next it overlaps with psychology since it includes behavioral study of elderly people and lastly it includes commercialization and therefore businessmen may be interested as well.

1.5.1 Elderly People

Elderly people require assistance and support to perform their daily activities of living. Our project will focus to provide nursing to these people so they can live independently in their homes and also their quality of life improves.

1.5.2 Patients

Patients having chronic diseases require support more frequently. They need special care and assistance. Our project aims to help health care sectors to monitor their patients efficiently and effectively.

1.5.3 Hospitals

There is a lot of patients in hospitals and it requires a great number of nurses and doctors to monitor them all the time. our project solves this problem by providing monitoring and nursing services to these hospitals[14].

1.5.4 Researchers

Researchers could be interested in the new techniques used to achieve the desired goals. They could also be interested in the results achieved to understand how different methodologies work on a variety of data sets. With such results and understanding available, further work can be done in the field to provide new methodologies and achieve better results.

1.5.5 Businessmen

Other than researchers, businessmen could also be interested in this project. The resulting developed product can be commercialized to be used for elderly home nursing. The investment in this product can help to get profits at an industrial level. Therefore this potential investment and possible market opportunities could interest many businessmen in this project.

1.5.6 Health Care Professionals

Health Care Professionals could also be interested in this project as it can nursing to elderly to improve their quality of life. Hence, it can improve and enhance their healthcare services.

1.5.7 Psychologists

Another group likely to be interested in this project could be psychologists studying the behavior of the elderly. This project could help them in acquiring important findings for continuing future research work. Sleep patterns can be used to identify and understand REM in the elderly population and provide an understanding of how sleep pattern changes under different circumstances. Dementia can also be understood in a comprehensive way along with its different steps. Early-onset signs of dementia can also be detected which could contribute significantly to the academic society of psychologists.

1.6 System Requirements

1.6.1 Hardware Requirements

- Processor: Intel Pentium 4 or later
- Operating system: Linux or Windows 8
- Memory: 4 GB minimum, 8 GB recommended
- Screen resolution: 1280x1024 or larger
- Internet connection: Required
- GPU: typically a Nvidia P100 (Pascal)

1.6.2 Software Requirements

- VS Code
- Jupyter
- Microsoft Project Professional
- Anaconda

1.7 Feasibility Analysis

Feasibility analysis is used to define whether the proposed project is feasible in terms of technology, resources, operations, and time frame for the development.

1.7.1 Technical Feasibility

The project can be implemented by using cameras and a web application. The data requirements can be fulfilled by using a camera for data set collection. In this case, the technology required is readily available in the market. The computing power is also easy to acquire with the advancement in technology.

For practical implementation, an internet connection and a computing device is required such as a laptop or a personal computer. Internet connection is needed as well. Graphics is also an important component for which a GPU is required so that computations can be done on images and videos.

1.7.2 Operational Feasibility

This project is most likely to be used by those in charge of elderly care as well as elderly people living alone. The Elderly can use it to get help for daily routine tasks whereas the assistance providers can use it for monitoring the behaviors and routinely tasks of those living alone. There can be resistance regarding privacy if cameras are used. This can happen in case a person does not want to be seen all the time directly or feels that such monitoring is evasive and invading. In such a case, it could be difficult to implement the project practically. People who are not resistant to using cameras are more likely to use this project.

1.7.3 Economical Feasibility

This project requires cameras for image acquisition. It also requires a computer with high computational powers for developmental purposes. The benefits of this project outweigh the costs because once implemented, it provides essential care for the elderly. Relocation in similar cases has much higher costs in the long term.

1.8 Challenges and Limitations

The main challenge is in the practical deployment of the project in such a way that it stays non-evasive while achieving the essential goals of smart home nursing and creating a home environment that is safe and secure. Acceptance of Smart Home Nursing by potential users is a major factor. The challenge is the data privacy and overall trust in the smart home services due to the inherited conservative behavior of elderly people towards technological devices. There is also a cultural constraint

in many countries regarding nursing homes which can cause hesitance. Another challenge is the installation of cameras in the home of the subject which requires professional assistance in most cases. This can be difficult to arrange for an elderly person by themselves.

There are many other challenges in the project. Some of the challenges are described below:

1.8.1 View-Point Variation

FIGURE 1.1: View-Point Variation

View-point variation occurs when an object is viewed from different perspectives. This can be a challenge if the person is seen from different angles by either change in the camera's position or a change in the person's position as shown in Figure 1.1. In such a case, the computer must identify all different views of the same person. In the example given below, the elderly person is facing the other side which can make it hard to identify the action as well as the person in the image[15].

1.8.2 Occlusion

When an object is not completely visible as in Figure 1.2 or is hidden behind another object. Occlusion in image processing makes it difficult to recognize objects and track them[16].

FIGURE 1.2: Occlusion

1.8.3 Illumination

Changes in illumination can also be a challenge since it alters the intensity level of images. It also alters how an image is perceived under different light conditions for example as shown in Figure 1.3. This can be an issue while processing the same object in daylight vs night light[17].

FIGURE 1.3: Illumination

1.8.4 Background Clutter

When there are a lot of objects in the background, it becomes hard to find the desired object. Because of the cluster of items in the backdrop, such as at a stadium or a busy venue with a lot of people, such photographs are described as noisy[18].

1.8.5 Data collection

One of the difficult challenges in this project is the collection of the datasets[19]. It is difficult to acquire consistent and quality data due to inconsistent data collection standards. Currently, there are many different international standards that can be used. However, the standards themselves are not consistent[20]. It may also be difficult to recruit participants for data collection. One of the reasons for this hesitancy is the invasion of privacy. Some other challenges in dataset collection are:

- Complicated dataset collection methods
- Complex instructions for elderly
- Literacy comprehension barriers
- Lack of understanding of the context
- Necessary approvals

1.8.6 Camera Synchronization

Another important issue is the synchronization of cameras to work simultaneously. IP surveillance must be synced in time by internal clocks either manually or automatically[21].

1.8.7 Limitations

A limitation in this project is in the availability of the helping person in case of an emergency and the limitation in how they can be contacted. Another limitation is concerned with the cameras used. The view or perception available at the moment by the visual input or cameras can be disrupted since an obstacle may hinder the input[22]. There is also a bias problem with the dataset in case it is imbalanced such as if it consists of more men than women. In such a case, the model will work differently for minor groups than for major groups[23].

One other major limitation is the lack of interoperability with other applications other than healthcare. The reason for this is the specificity of the area of healthcare and its lack of overlap with other fields. The geographical spread of the elderly subjects is another project restriction. The subjects are all from Pakistan. More older individuals from throughout the world, on the other hand, maybe added to cope with a wide range of major variations in opinion.

Chapter 2

Literature Survey

2.1 Literature Review

'=

Using a hybrid approach, Jin-Hyuk Hong, Julian Ramos, Choonsung Shin, and Anind K. Dey[24] proposed a real-time recognition system for everyday living activities. For the well-being of older individuals in various worrying conditions such as dementia, fits, and Alzheimer's patients, activity recognition for ambient assisted living is required. Walking, standing, sitting, lying down, bending, falling, and biking are the seven actions identified in the suggested strategy. The use of physiological sensors only reveals half of the tale; it does not indicate that the individual is participating in physical activity. As a result, a hybrid strategy is presented to characterize the link between sensory inputs and activity using Bayesian networks and support vector machines.The system comprises a Bioharness BT physiological signal recording device, a unique indoor/outdoor localization system, and a laptop/smartphone for data logging. Modeling, calibration, and online recognition are the three steps of the system.

We create a pool of activity models from a group of people during the modeling stage. 3D acceleration data from the bioharness is acquired using an android smartphone, which records videos for manual data categorization and the construction of a model using Bayesian networks and support vector machines[25]. The recognition performance of activity models on new data is calculated during the calibration phase, and the most accurate model is chosen. The activity recognition is carried out by the hybrid model that was chosen during the calibration phase, and it is refined further by many criteria of localisation. For validation, 15 people wearing Bioharness in a furnished room performing various jobs for an

hour are videotaped and data is collected.The proposed system is compared to two different activity model building approaches: population and individual. When compared to the other two ways, the proposed methodology has the best accuracy (74%) of the three. Standing vs. walking (8.7% mistakes overall) and standing vs. sitting (8.7% errors total) were difficult to classify in the proposed technique due to their closeness in terms of chest movement (3.4% percent errors in total).

Sandipan Pal and Charith Abhayaratne[26] developed a video-based new paradigm for recognising activity levels in smart homes. Aging is a worldwide phenomenon, with the world's old population growing at an alarming rate. The demand for smarter houses with the newest technologies for assisted living is increasing in order to support healthy and independent living for the aging population. The experiment is done using two sets, one with a single camera and the other with a dual-camera system, according to the given technique. Optical flow is used to extract motion data from the input video, which are subsequently fed into a neural network for activity level classification[27]. An individual's activity level is a reflection of the amount of time they spend doing everyday activities. As proposed, there are three levels of activities: High Activity Level indicates physical movement, Low Activity Level indicates minimal or no movement and No Activity Level indicates no movement. For a performance comparison under a single camera as well as an orthogonally positioned twin camera setup, two distinct feature vectors with non-intrusive motion features are employed. Single camera setups utilising basic motion features are good, but performance varies when changing camera location. The performance of a dual camera arrangement is better than a single camera setup since the features of both cameras are fused. The neural networks' results reveal that motion characteristics may be utilised to accurately evaluate individual activity levels.

Rainer Planinc (2016)[28] discussed the relevance of computer vision methods in Ambient Assisted Living (AAL), how cameras are used for this, and what the current state-of-the-art is in terms of usage and recognition algorithms. With the goal of introducing computer vision to specialists in other AAL domains in general and its applications to assisted living in particular, he distinguished between basic RGB cameras and depth sensors. From different cameras that may be used in care facilities and people's homes to fresh developments in video and image feature extraction and classification, the image processing pipeline[29] has been studied. The most successful data formats and comprehensive analysis of depth sensors for feature estimation algorithms were also covered. Although the use of RGB cameras to monitor elderly persons create privacy issues, computer vision is

widely employed in AAL. The reason[30] for this is that there are a variety of AAL scenarios in which the use of cameras is still permissible, including nursing homes and hospitals, as well as certain events or activities such as safety assessments. Video and picture may also provide a wealth of information about a person's activities. As a result, cameras provide rich sensor data for human monitoring, not only supplementing but also soon replacing systems with networks of binary sensors. Human behaviour analysis, Fall detection, Tele-rehabilitation[31], Gait analysis, and Physiological monitoring are some of the video camera applications suggested for AAL situations. The image processing phases were picture capture, image pre-processing, feature extraction, and recognition. Furthermore, the benefits of depth sensors were emphasised, such as the lack of a separate light source and the ability to apply conventional algorithms to depth data. The versatility and flexibility of vision-based approaches are their key advantages.

Marco Buzzelli (2020)[32] presented a monitoring system that uses computer vision algorithms and visual sensors such as RGB cameras to provide help to elderly persons in their homes[33]. These technologies allow a person's activities of daily living (ADL) to be monitored and aberrant patterns in activities to be detected. A detailed review of current video datasets (UCF Sport, Hollywood 2, UCF-101, Kinetics, Charades, NTU) for action identification revealed that no dataset is regarded suitable in terms of classes. ALMOND (Assisted Living MONitoring Dataset), a merged action dataset of 6775 training and 790 test sets, was employed as a training set for a vision-based monitoring approach. A single-camera situation is utilised to monitor a person's actions, with the system being combined with powerful action recognition algorithms for monitoring interior settings. Furthermore, the monitoring system sends out alarm signals when abnormal occurrences occur. A user-friendly interface with several applications was created, including raw data collecting, visual log production, activity monitoring, and anomaly inference support.

In terms of recognition accuracy, the suggested method is tested utilising a variety of state-of-the-art architectures, including Two streams (RGB+OF), C3D+Linear SVM, VideoLSTM, and DeepHAR (RGB alone). 97% accuracy on fundamental posture inference, 83% accuracy on alerting situations, and 71% accuracy on activities of daily life. The maximum allowable distance between the camera and the monitored individual is likewise approximated using a broad technique. Finally, defined actions and the trained model are integrated into a computer vision-based application designed specifically to monitor elderly people in their homes.

Nicola Mosca[34] and colleagues presented a network of low-cost RGB-D sensors with no overlapping fields-of-view for detecting aberrant human behaviour in surveillance and AAL applications. Population aging is expected to become a major worldwide problem in the near future, and it is already posing a significant burden in several nations. As a result, the demand for assisted living facilities for older persons who want to live independently has soared. In ambient assisted living, automatically detecting irregularities in human behavior is an important technique. The suggested technology extracts torso nodes from the OpenNi framework's skeletal characteristics and use the Kalman filter for people tracking. Finally, abnormal behavior is detected using a variety of classification algorithms and approaches, including ANN, SVM, and k-NN.
In an experiment, a set of RGB-D cameras with three Kinect sensors were deployed in an interior area by a group of researchers. The Kalman filter is used to detect and track user skeletons using the OpenNI framework and the Primesense NiTE library. Three distinct classifiers employ trajectory data to determine whether the selected user's behavior is abnormal. This experiment used three distinct supervised classifiers: a support vector machine (SVM), a k-nearest neighbour (k-NN), and a neural network (NN). The results demonstrate that SVM and neural networks may achieve accuracy levels of over 90%, whereas k-NN has the lowest performance. The top classifier was an overall neural network, which had a 93.3% accuracy. The proposed architecture and developed methodologies can recognise anomalous behavior in the majority of cases involving the total of observed path, but the system may fail if multiple people enter the camera field of view at the same time and do not maintain the same walking direction while crossing occluded areas[35].

In industrialised countries, integrating fragile individuals into society is a serious concern. Continuous monitoring of fragile persons at home for the detection of worrying circumstances can be a strong instrument for their inclusion in society by allowing them to live freely while being safe. Bruna Maria Vittoria Guerra[36] and others proposed a pose recognition system for disabled students based on skeletal tracking through four Kinect One devices independently recording the inhabitant from different viewpoints while maintaining the individual's privacy. The suggested method is based on Microsoft's Kinect One motion sensing device, which can provide 3D coordinates of skeletal joints of interest. An RGB camera (1920 x 1080 pixels), a depth sensor (512 x 424 pixels), and an array of four microphones are used in the Kinect One motion-sensing system to identify a human body and speech input (48 kHz). Two Kinect One devices are used to detect the entire room, while the other two are used to acquire two particular parts of the room,

such as the bed and the desk. The major focus was on the three most common poses: standing, sitting, and lying down, with one additional position, "hazardous sitting," which collected all cases of malaise or fainting that resulted in a seated individual slumped or lying backward. 15 normal participants' datasets were collected to train neural networks. After that, the angles and joint position traces from Kinect One devices were visually evaluated alongside a graphical depiction of the rebuilt skeleton to assign them to one of the four postures listed above. A network is made up of three layers of neurons that are fully linked, as well as an input layer that is connected to the ten attributes that describe each frame in the database. To investigate the MLP network's classification resilience, it was trained and tested 50 times. With an average accuracy of 83.9%, the suggested method produces good results. However, due to certain misclassification mistakes, the classifications of sitting and unsafe sitting have a lower accuracy.

Philipe A. Dias (2020)[37] emphasised the need for supported living for the elderly and presented a strategy in which gaze direction is one of the most powerful and crucial indicators of a person's involvement with the environment. Gaze estimation is employed in this strategy to detect reciprocal interactions between items and their users. The major purpose was to aid doctors by providing them with reports on patients' mobility and IADLs (IADL). To calculate gaze estimate, the proposed method exclusively depends on the positions of face keypoints, which are extracted using Openpose. He suggested a simple Neural Network Regressor to assess the gaze direction of people utilising multi-camera assisted living situations. In addition, to deal with occurrences of facial keypoint occlusion, the model uses a new confidence gated unit in its input layer. Furthermore, to account for the fact that gaze estimation changes depending on the context, the model generates an estimate of its own prediction uncertainty.

Eva-Maria Schomakers and Martina Ziefle want to learn more about privacy problems in AAL and how the features of AAL technology and systems affect privacy perceptions. Ambient Assisted Living (AAL) technology enable seniors to age in situ (Peek et al., 2014). AAL has the ability to assist health-care systems deal with the enormous issues that the ageing population faces. Despite their promising potential and ever-growing product variety, demand for and spread of AAL technologies is surprisingly modest, and prospective user acceptability is a crucial reason. Acceptance is influenced by the target group of AAL (the elderly). A questionnaire technique with Maximum Difference Scaling (Max Diff) questions was used to investigate privacy concerns and how system parameters impact users'

perceptions of privacy. Three focus group meetings were held as part of the pre-study, during which the participants explored the hurdles and benefits of AAL in general, as well as their worries about privacy in AAL contexts. According to the studies, the most common privacy issue is data abuse, but there are also worries about perceptions of monitoring and technology intrusiveness. The sole issue expressed by the participants is that they are concerned about the automatic decisions made by AAL technologies concerning them. The MaxDiff study clearly demonstrates that the data recipient is the most critical system attribute in terms of privacy. Users want to choose who they trust with their personal information.

Pau Climent-Pérez (2019)[38] analysed all of the most recent and groundbreaking breakthroughs in the domains of Ambient Assisted Living (AAL), computer vision, intelligent systems, and lifelogging. This research reviewed all prior strategies and approaches in these sectors, as well as analysing contemporary intelligent techniques used with various video-based lifelogging technologies in order to propose AAL lifelogging services. Furthermore, these technologies' privacy problems and ethical difficulties are examined. The primary goal of this study is to improve understanding of the many and varied disciplines involved in ambient assisted living. The breadth of the preceding strategies is limited in several ways. The four kinds of machine learning approaches described are spatio-temporal networks, deep generative networks, multiple stream networks, and temporal coherency networks.

According to Daniel Flores-Martin (2021) citeS1, the world's population is ageing, and younger people are migrating to wealthier places. Nursing homes are in higher demand as more older individuals seek to live independently. He presented an architecture that would make it easier to integrate and collaborate IoT devices from various manufacturers. Furthermore, the proposed architecture made use of machine learning (ML) approaches to increase the identification of old people's activities in nursing facilities in rural and sparsely inhabited areas. In smart homes, machine learning algorithms and IoT devices are used to monitor and identify older people's behavioural patterns, as well as make predictions for future occurrences. The design uses data from sensors and cellphones to adjust the behaviour of the environment's actuators. It also uses this information to learn from the environment and forecast actions that should be taken based on previous actions. A use case based on a nursing home in a rural region is used to implement the architecture. Multilayer Perceptron Neural Networks are used in conjunction with LSTMs to guarantee that the model is aware of time series data and can properly anticipate human behaviour over time. The elderly's quality of life is improved in a simple, economical, and transparent manner using this strategy.

This design, in particular, may be used to any nursing home that wishes to monitor its patients using the most up-to-date technologies at a reasonable cost, regardless of the technology utilised.

2.2 Motivation

There has been an exponential increase in the old age population recently. This increase can be attributed to the development of medical expertise and a healthy lifestyle. With this increase comes the responsibility of taking care of the elderly as most people at this age require constant monitoring in case of any emergency and to help them perform daily tasks. This goal can either be achieved by hiring nurses, staying at home, or institutionalizing the elderly. All of these options are costly and take away the privacy of older people. We aim to provide a cost-efficient, economic solution to monitor elderly people and notify guardians in case of any emergency. This will not lead to economic growth but also help to maintain the privacy of elderly people.

2.3 Problem Statement

To develop a system for monitoring daily activities and provide risk analysis for the elderly to better their standard of living and give them an independent lifestyle. The main idea of Smart Home Nursing is to investigate the use of multiple cameras for the detection, recording of daily activities and lifestyle patterns as well as the development of intelligent mechanisms for translating visual input into accurate situational assessment and rapid response.

2.4 Our Contribution

Current state-of-art methods are making use of sensory data to develop novel technologies for providing smart home nursing[39]. In this project, visual input such as images and videos is used for training algorithms and understanding activity patterns. The main goal is to improve the current methodologies employed in providing assisted living for elderly[40].

Activities of the elderly are monitored remotely to detect anomalies and generate alarm so that timely action can be taken. Daily activities are used to detect immediate dangers and the same activities are used to create a record. The record can be used later to compute a long-term analysis[2]. Our system aims to provide detection of a multitude of significant activities which could provide help to provide assistance for living alone.We aim to create a system for detecting emergency situation and providing a record for behaviour analysis.

TABLE 2.1: Comparison of state-of-art work

Author	**Methodology**	**Dataset**	**Activities**	**Pros**	**Cons**
Daniel Flores Martin, 2021 [5]	IoT and ML techniques	Sensors	ADL	scalability	N/A
Marco Buzzelli, 2020 [32]	DL,R-CNN,I3D and Deep-Har	ALM OND	Seating, walking lying standing	User-friendly system	Lack of usability
Philipe A. Dias, 2020 [37]	Openpose, NN Regressor	Gaze Follow	Video streams (IADL)	object segmentation and gaze estimation	N/A
Bruna Maria Vittoria Guerra, 2020 [41]	standing, sitting, lying down and dangerous sitting	Sensors	Dangerous sitting which grouped all situations of fainting resulting	Preserve the privacy	Accuracy for some activities is relatively low
Nicola Mosca, 2018 [34]	OpenNI framework,NN,SVM and k-NN	videos and sensors	Walking behaviour	93.3 accuracy % for NN anomaly detection	Fail for multiple people
Rainer Planinc, 2016 [28]	videos and sensors	videos and sensors	Fall detection, Gait analysis, Pose analysis	flexibility and adaptability	Privacy

TABLE 2.1: Comparison of state-of-art work

Author	**Methodology**	**Dataset**	**Activities**	**Pros**	**Cons**
Sandipan Pal, 2015 [26]	Motion features extraction and then send to NN for classification	cameras	High Level Activities, Low Level Activities and No Level Activities	Provide benchmark for video-based assisted living research	View-Point Variation
Jin-Hyuk Hong, 2013 [24]	Bayesian networks and SVM for activity recognition	Sensors and videos	walking, standing, sitting, lying ,bending,falling and bicycling	Improved accuracy	Difficulty in classifying some activities

Chapter 3

Dataset

3.1 Existing Datasets

3.1.1 Kinetics 400

It's a video dataset for classifying human actions. The inspiration for this dataset is ImageNet, this dataset is the successor to HMDB-51 and UCF-101 which are both human action video datasets. This dataset contains clips that are only 10s long. Clips can be utilised for multi-modal analysis because they have sound. This dataset is more focused on human actions rather than activities. The actions included are Person Actions (Singular) such as drinking, Hugging and washing dishes are examples of person-to-person and person-to-object actions. This dataset has 400 action classes where each class contains 400-1150 instances (clips). The total instances in the dataset are 306,245. A clip[42] may contain multiple activities such as "texting" while "Walking" but that clip will only come under one of the class labels. The data collection process is simple, firstly class list for human actions is decided using multiple sources such as past action datasets, motion capture datasets and finally, crowdsourcing systems such as Amazon Mechanical Turk (AMT) can be used. After that videos from YouTube are collected by matching titles with action lists made. These videos are not professionally shot. Image classifiers are used on the videos that extract the clips where some activity is being performed and each clip comes from a separate YouTube video. Next people from AMT are used to label the clips and identify whether the supposed action is occurring during the clip or not. If a clip obtains 3 out of 5 positive responses then it is included in the dataset. The dataset gets cleaned up in the last stage. First of all the dataset is de-duplicated by taking the feature vector from the sampled frame of each video and checking similarities between vectors. After that noisy

classes are detected and resolved by either merging or removing classes. Finally, final filtering is performed by sorting examples from most to least confident. The dataset file contains a label that indicates activity performed, youtube_id which is a video identifier on YouTube, time_start and time_end (in seconds) to indicate the section within a video where activity is performed and split to tell if it is testing, training or validating data. Dataset is split such as 246245 clips are utilised for training, 20000 for validation, and 40000 for testing.

3.1.2 ActivityNet

Many available datasets are very simple. The ActivityNet dataset is a large scale video-based dataset with the goal of covering a wide range of complicated activities[43]. There are 203 activity classes in the dataset, each with an average of 137 untrimmed films. Each video has 1.41 activity instances and there are a total of 27801 videos with 849 video hours. It's based on a semantic ontology, and it's the first human activity database to classify a significant number of videos into a semantic taxonomy. Because it is difficult to define a semantic organisation for activities, this dataset employs the Department of Labor's activity taxonomy developed for the American Time Use Survey. This taxonomy has over 2000 activities and for this dataset subset of 203 activity categories is manually selected. Personal Care, Eating and Drinking, Household, Caring and Helping, Working, Socializing and Leisure, and Sports and Exercises are the seven top-level categories in which these 203 activity classes fall. Data is collected in the following steps first of all videos are collected from YouTube by searching the activity labels. Videos are verified with the help of Amazon Mechanical Turkers who view each video and determine if it contains the intended activity and if not then the video is removed. To maintain the quality only expert Turkers are employed. This gives verified untrimmed videos. Then the specific portion of the video where desired activity is performed is annotated manually. This is also done by Turkers. One video may contain more than one activity label. The dataset file contains the label that indicates activity performed, duration to tell the total video time, segment with start time and end time (in seconds) to indicate the section within a video where activity is performed, subset to tell if it is testing, training, or validating data and a URL which is a video link on YouTube. 50% of data is used for training while 25% is used for validation and testing.

3.1.3 Kinetics 600

It is an extension of Kinetics-400. The main purpose of the kinetics project was to replicate size of ImageNet which has 1000 classes so this dataset moves toward

the goal by increasing 400 classes to 600 classes and increasing the number of clips from 300,000 to 500,000. The data collection process is similar to the Kinetics 400[44]. A bit difference is that instead of only collecting classes from existing datasets, Google's Knowledge graph and YouTube autocomplete are also used. Data is collected by querying YouTube in two languages: English and Portuguese as these two languages have the most native speakers in the world. Dataset is split such as 392,622 clips are used for training, 30000 for validation, and 60000 for testing.

3.1.4 Kinetics 700

It is an extension of Kinetics-400 and Kinetics-600 just like them it is a dataset of video clips. Data is collected from YouTube.There are 700 action classes in this dataset, each with at least 700 video clips.This dataset[45] collects new clips for 123 rarest classes that contain less than 700 clips. To get more clips for rarest classes search on YouTube is done by various means like using synonyms or searching in different languages i.e. English, French, Spanish and Portuguese. In this way, more clips can be obtained. The same clip can occur multiple times so to filter these out clips are clustered and then by looking at individual clusters duplicates are removed. Same as kinetics 400 the dataset file contains a label that indicates activity performed, youtube_id which is a video identifier on YouTube, time_start and time_end (in seconds) to indicate the section within a video where activity is performed and split to tell if it is testing, training or validating data. Dataset is split such as 545793 clips are used for training, 35000 for validation, and 70000 for testing.

3.1.5 AVA-Kinetics

It is a video dataset of human action. This dataset is made by the crossover of the AVA dataset and Kinetic-700 dataset by doing AVA style annotation on the Kinetic-700 dataset. AVA dataset has 80 actions and 430 video clips each 15 minutes long.AVA dataset[46] is expensive to annotate as it annotates actions for each human in a video clip on each frame captured after one second. AVA dataset also produces a bounding box around each person and each person has all the co-occurring activities labels. Kinetic-700 has over 700 classes with each class having at least 700 clips. AVA-Kinetic dataset annotate the single frame of each video clip of the Kinetic dataset with bounding boxes and atomic actions. To do this first of all person is detected on each frame using a pre-trained Faster RCNN, then a frame with the highest person detection is chosen from each video clip. After this human annotators verify data for any missing bounding boxes and in case of annotate it.

A 2 second video centered on the keyframe is obtained and many human raters (at least 3) propose action labels for the person in the keyframe. At the end labels are verified by at least 2 to 3 raters. For annotation 27 AVA classes with poor recognition are selected and 115 relevant classes are picked from kinetic AVA class is applied to all kinetics videos which match with the actions. The dataset file contains video_id which is a video identifier on YouTube, middle_frame_timestamp in seconds from the start of the YouTube video to indicate a particular frame, person_box displays bounding box of a person (x1, y1) and (x2, y2), action_id to give the id of action performed and person_id which is optional for kinetic dataset files it links a bounding box to others if depicting the same person. Dataset is split such as 141,475 clips are used for training, 32,592 for validation and 64,902 for testing.

3.1.6 HACS (MIT)

Human Action Clips and Segments (HACS) is a video dataset for recognising human action. The dataset has 200 action classes similar to ActivityNet. The dataset employs 200 labels to query YouTube, which returns 890k videos with videos per class ranging from 1100 to 6600, after deleting duplicates, the dataset contains 504,000 films with an average length of 2.6 minutes. This dataset has two types of manual annotations first is action labels on 1.5M clips of 2 seconds that are retrieved from the videos. These clips are identified by comparing the results of several visual classifiers and 0.6M clips are annotated as positive while 0.9M clips are annotated as negative samples. This dataset is called HACS Clips.The second[47] annotation type identifies temporal localization from 50,000 untrimmed videos and label them. This dataset is called HACS Segments and each video has at least one action segment. The segments in this dataset are shorter, 4.7x more segments and 2.5x more videos as compared to ActivityNet. The dataset file contains class name that indicates activity performed, youtube_id which is a video identifier on YouTube, subset to tell if it is testing, training or validating data, start and end (in seconds) to indicate the section within a video where activity is performed and label which if 1 means positive same and if -1 means negative sample. Authors split the collection into training, validation and testing sets of size 1.4M, 50K and 50K clips, which are sampled from 492K, 6K and 6K videos, respectively.

3.1.7 MMACT

To tackle the disadvantages of vision-based modalities, MMACT is a recent large-scale dataset. Its main goal is to detect and recognize the action tasks and

cross-modal action study. There are 36000+ temporally action instances in this dataset[48] for 37 action classes, spanning a wide range of daily living tasks like as desktop-related and check-in-based stuff in four different settings. There are 7 types of data modalities. 20 subjects are involved in this research who were wearing smart glasses to record egocentric videos. Smartphones installed with some initial sensors were carried and placed inside the subject's pants to gather data. They were also wearing smartwatches. 4 scenes were designed inside indoor environments which were: free space, occlusion, entrance, and desk work. On cross-subject, scene, view, and session evaluation criteria, the studies on this dataset have proven that it is quite effective.

3.1.8 Home Action Genome

It is a large-scale dataset used for action recognition for a multi-view video databases for indoor activities. Data is collected[49] from diverse viewpoints for both low-level and high-level action definitions. 27 participants were recorded at different places in two different houses. There were 27 sensor types used including cameras (RGB), infrared (IR), microphone, RGB light, light, acceleration, gyro, human presence, magnet, air pressure, humidity, and temperature. These sensors were attached at different places in the houses and on the heads of the participants. This dataset includes (1) video-level activity labels, (2) temporally localized atomic activity labels, and (3) spatio-temporal scene-graph labels. There are 75 annotated activities.

3.1.9 SoccerNet

A Scalable Dataset for Action Spotting in Soccer Videos. This contains is 500 complete soccer games, every game is composed of 2 untrimmed videos, 1 for each half period[50], from six main European leagues, covering three seasons from 2014 to 2017 and a total duration of 764 hours. A total of 6,637 temporal annotations are automatically parsed from online match reports at a 1-minute resolution for three main classes of events (Goal, Yellow/Red Card, and Substitution). As such, the dataset is easily scalable. These annotations are manually refined to a one second resolution by anchoring them at a single timestamp following well-defined soccer rules. This dataset focuses on the difficulty of localising relatively sparse events inside long videos, with an average of one event per 6.9 minutes.

3.1.10 VidSitu

VidSitu is a dataset for the task of semantic role labeling in videos (VidSRL). It is a large-scale video[51] understanding data source with 29K 10-second movie

clips richly annotated with a verb and semantic-roles every 2 seconds. Entities are co-referenced across events within a movie clip and events are connected to each other via event-event relations. VidSitu clips are selected from a huge (3K) collection of videos and are designed to be both complicated (4.2 unique verbs each video) and diversified (200 verbs have over 100 annotations each).

3.1.11 Lifelog Dataset

The NTCIR-12 Lifelog dataset was produced as a test baseline to help the Information Retrieval (IR) community develop new and novel lifelogging retrieval and display techniques. The NTCIR Lifelog test collection consists of data from three life loggers for about one month each. The data consists[52] of a large collection of wearable camera images (at about 2 per minute) and an XML description of the semantic locations (e.g. Starbucks cafe, McDonald's restaurant, home, work) and the physical activities of the user (e.g. walking, transport, cycling), of the life logger at a granularity of one minute. Because lifelog data is often visual, the output of the CAFFE CNN-based visual concept detector was included in the test collection as supplementary metadata to lower the barriers to participation. For each image, this classifier generated labels and probabilities of occurrence for 1,000 objects. The CAFFE visual concept detector's accuracy is comparable to that of today's off-the-shelf visual analytics solutions. All faces in the photographs were blurred to anonymize the data, and only semantic locations were provided.

3.1.12 IKEA ASM

This is a multi-modal and multi-view video dataset of assembly tasks to enable rich analysis and understanding of human activities through Actions, Objects and Pose[53]. It contains 371 samples of furniture assemblies and their ground-truth annotations. Each sample includes 3 RGB views, one depth stream, atomic actions, human poses, object segments, object tracking, and extrinsic camera calibration. The 48 participants put together furniture in five distinct offices, labs, and houses. Three Kinect V2 cameras made up the data collection hardware equipment. These three cameras are positioned to capture images of the work area from the front, side, and top. There are 16,764 annotated activities in the dataset, with an average of 150 frames per action (6sec).

3.1.13 WISDM

The "WISDM Smartphone and Smartwatch Activity and Biometrics Dataset" contain data gathered from 51 subjects, who were asked to perform 18 different tasks for 3 minutes each [54] Smart watches and smart phones were used for data

gathering. The data was collected in controlled labs. Total of 18 activities were performed by the subjects such as walking, jogging, stairs, sitting, standing, typing, clapping, eating pasta, etc. These activities were divided into three categories: 1- Activities that do not require the use of one's hands 2- activities focused on the hands (General) 3- activities focused on the hands (Eating). The accelerometer and gyroscope on both the phone and the watch record raw time-series sensor data at a rate of 20Hz.

3.1.14 Opportunity Dataset

Opportunity dataset is a dataset designed to create different activity recognition algorithms from data collected from motion sensors. The dataset has a total of 2551 instances with 242 attributes. This dataset has a large number of activity instances and sensors of different modalities which simulates opportunistic data in a real environment. This dataset[55] is used in many other areas as well like Classification, Automation Segmenting, Sensor Selection, and Features selection Dataset collected data from three different types of sensors:

- Body-worn Sensors
- Object Sensors
- Ambient Sensors

Body-worn sensors have 145 attributes, Object sensors have 60 attributes and Ambient sensors have 37 attributes. Activity recognition environment with 73 sensors having 10 modalities are deployed on body, in the environment and on objects. Subjects carry out activities in a room simulated with a deckchair, kitchen, door to go outside, coffee machine, chair, and table. People perform activities of daily living in this environment, and data from this environment is synchronized to annotate with video footage. Four users are operated in this environment each performing ADL 'run' in which there is natural execution of activities 5 times and one 'drill run' in which users perform scripted activities. Finally, 25 hours of data are recorded and classes are done on five tracks Locomotion which indicates activities like sitting, and standing, HL_Activity which define high-level activities, LL_Left_Arm, and LL_Right_Arm which shows left and right arm movements, ML_Both_Arms which define activities involving both arms movement.

3.1.15 UCI-HAR Dataset

UCI-HAR dataset is created by recording subjects wearing waist-mounted smart phones embedded with two sensors performing Activities of daily Living in a real

environment. 30 people[56] with an age limit of 19-48 years participated in the experiment. Every person is carry out 6 activities:

1. WALKING
2. WALKING_UPSTAIRS
3. WALKING_DOWNSTAIRS
4. SITTING
5. STANDING
6. LAYING

Each person is wearing a smartphone embedded with accelerometer and gyroscope sensors on the waist. These two sensors recorded linear acceleration and angular velocity at a rate of 50HZ. A video of subjects performing 6 activities is recorded and features are generated from the data collected from the sensors with the 6 activity classes mentioned above. The features selected for this dataset come from accelerometer and gyroscope sensors tAcc-XYZ and tGyro-XYZ. Acceleration signal is separated into body gravity acceleration signals by applying ButterWorth Filter and jerk signals are obtained (tBodyAccJerk-XYZ and tBodyGyroJerk-XYZ). Subsequently other filters like Euclidean norm, Fast Fourier Transform is applied to generate tBodyAccMag, tGravityAccMag, tBodyAccJerkMag, tBodyGyroMag, tBodyGyroJerkMag.

3.1.16 USC-HAD Dataset

UCS-HAD dataset used a standard to compare different algorithms in scenarios of health care. UCS-HAD dataset helps to build a powerful recognition system that requires training on the large group of people with diversity. 14 subjects[57] 7 male and 7 female with age 21-49, height 160-185 cm and weight 43-80 kg are participated in experiment to collect data. 12 different activities are used in this experiment mentioned below:

1. Running Forward
2. Jumping
3. Sitting
4. Standing

5. Walking Forward
6. Walking Left
7. Walking Right
8. Walking Upstairs
9. Walking Downstairs
10. Sleeping
11. Elevator Up
12. Elevator Down

Each subject wears a mobile phone pouch at one's front right hip which is embedded with MotionNode which is a motion sensing application and sends data to a laptop via cable. And each subject hold a miniature in one hand and perform 5 trials of activities on different days. This experiment is recorded and segmented based on starting and endpoints recorded by the observer. Finally, each activity trial is stored in a separate file. Each .mat file contains height, weight, date, serial number, activity name, activity number, sensor location, orientation, and readings.

3.1.17 Skoda Dataset

Skoda mini dataset describes the activities of assembly line workers in a scenario of car production. In the experiment[58] a person is recorded wearing a 20 3D accelerometer with 60 attributes on both hands to recognize both arms' motion in a car maintenance scenario. Each accelerometer used in this experiment depends on its values on the x, y, and z-axis. The subject performs 10 activities. The list of activity classes are :

1. Holds a notepad in their left hand and writes down a sentence with their right hand.
2. Open the hood with the left hand and blocks it with a stick kept in the right hand
3. Removes stick with his right while holding the hood in the left hand
4. Checks the gaps with his both arms
5. Grabs and closes the car door with the left hand

6. Grabs and close front and back with the left and right arms at the same time
7. Checking the gaps by moving both hands simultaneously
8. Open trunk with both hands
9. Grabs the steering wheel with both hands

3 hours of recording is recorded with 70 repetitions per gesture with 98HZ sampling data. The above experiments give 240 features per instance.

3.1.18 PAMAP2 Dataset

PAMAP2 Physical Activity Monitoring dataset is used in activity recognition and intensity estimation by applying data processing, segmentation, feature extraction, and classification. In PAMAP2[59] 9 subjects are participating in the experiment wearing 3 inertial measurement units and a heart rate monitor sensor. These 9 subjects perform 18 different activities like cycling, walking, playing soccer, etc. 1 IMU sensor is on the wrist of the subject, 1 IMU on the chest and 1 IMU on the dominant side ankle. HR monitor at a sampling frequency of 9 HZ. These subjects are recorded performing these 18 different activities.

3.1.19 DAPHNET Dataset

Freezing of gait (FOG) is one of the most incapacitating motor symptoms in Parkinson's disease (PD). The occurrence[60] of FOG reduces the patients' quality of life and leads to falls. Questionnaires are commonly used to assess FOG, but this method is subjective and may not accurately reflect the severity of this symptom. Sensor-based devices can give precise and objective data to follow the progression of symptoms and improve PD management and treatment. DAPHNET dataset is used as the standard for automatic algorithms to recognize the Freezing of gait. Dataset is recorded by generating many freezing events in the lab. 7 male Subjects with PD and with 62-70 age wearing wearable acceleration sensors on legs and hips performing three different activities:

1. Walking in straight lines
2. Walking with numerous turns
3. Real life ADL activities

Then this dataset is used in recognizing Freezing of Gait.

3.1.20 Rose NTU Dataset

Rapid Rich object Search lab has two datasets. We used a dataset with 60 action classes and 56,880 RGB video samples. The activities were in three major categories daily actions, mutual actions, and medical conditions[61].

Three Kinect V2 cameras capture each dataset at the same time. RGB videos have a size of 1920x1080 pixels, depth maps and infrared movies have a resolution of 512x424, and 3D skeletal data has the 3D coordinates of 25 body joints at each frame. Some examples of actions are:

1. staggering
2. falling down
3. headache
4. chest pain
5. drink water
6. Stand up
7. sit down
8. eat meal
9. open bottle

3.1.21 Toyota Smart Dataset

The Toyota Smarthome dataset was captured in a residence with seven Kinect v1 cameras. It comprises 18 subjects' common daily living activities. The participants are senior citizens between the ages of 60 and 80. The dataset contains a 640x480 resolution and three modalities: RGB, Depth, and 3D Skeleton. RGB was used to extract the 3D skeletal joints. The subjects' faces have been blurred for privacy concerns. The dataset is now available in two versions: Trimmed Toyota Smarthome vs Untrimmed Toyota Smarthome[62].

We use the Trimmed dataset version, in which we select 10 actions from a total of 31. The videos were clipped for each activity, resulting 16,115 short RGB+D video samples in total. Activities were carried out in a natural way. As a result, the dataset poses a unique combination of challenges: high intra-class variation, high-class imbalance, and activities with similar motion and high duration variance. Both coarse and fine-grained labels were used to label activities. These

characteristics differentiate Toyota Smarthome Trimmed from other datasets for activity classification[63]. These activities are:

1. Drinking
2. Eating
3. Enter
4. Laydown
5. Leave
6. Stand up
7. Sit down
8. Use telephone
9. walk
10. Take medicine

3.1.22 UR Fall Dataset

This dataset has 70 sequences (30 falls + 40 daily living activities). Falling events are captured using two Microsoft Kinect cameras and accelerometric data[64]. Only one device (camera 0) and an accelerometer are used to record ADL occurrences. PS Move (60Hz) and x-IMU (256Hz) devices were used to collect sensor data. The following is how the data is arranged. Each row contains a succession of depth and RGB images for cameras 0 and 1 (installed on the floor and ceiling, respectively), as well as synchronisation data and raw accelerometer data. Each video stream is saved as a png image sequence in a separate zip package. PNG16 format is used to hold depth data.

TABLE 3.1: Comparison of Existing Datasets

Dataset	**Instances**	**Modality**	**Classes**	**Source**	**Activities per Class**
Toyota Smarthome[62]	16,115	RGB + D videos	31	Toyota	1
Kinetics 400[42]	306,245	Videos	400	DeepMind	1
Activity Net[43]	(27801) 849 video hours	Videos	203	KAUST	1
Kinetics 700[45]	650000	Videos	700	DeepMind	1
AVA Kinetics[46]	238906	Videos	80	Google Inc	1
Kinetics 600[44]	500,000	Videos	600	DeepMind	1
HACS (MIT)[47]	1.5 M	Videos	200	MIT	1
MMACT[48]	36K average length from 3-8 sec	RGB videos, Key points, Gyroscope, Acceleration, Orientation, Wi-Fi and Pressure, Signal	37	Hong Kong University of Science and Technology	37
Home action genome[49]	1.7M human object relationship instances	Audios and Videos Light, RGB, Humidity,	86	Stanford University	18

TABLE 3.1: Comparison of Existing Datasets

Dataset	**Instances**	**Modality**	**Classes**	**Source**	**Activities per Class**
SoccerNet[50]	6637	500 online videos	3	KAUST, Saudia Arabia	Goal card substitution
VidSitu[51]	29K 10-sec movie clips	Movies, Videos, Clips	6	University of Southern California	Talk, Deflect, Leap, Punch
Lifelog[52]	88,124	Accelerometer, Temperature, and GPS	N/A	School of Computing, Dublin City University, Ireland	Personal lifelog data
IKEA ASM[53]	371	3 RGB Views	33	Australian National University	Human Poses
WISDM[54]	15630426	Accelerometer	6	CS Department, Fordham University.	Human Poses
Opportunity[55]	2551	Videos	5	ETH Zurich, EPFL	9
UCI -HAR[56]	10299	Videos	6	Non-Linear Complex Systems Laboratory	1
USC -HAD[57]	N/A	Videos	12	ACM International Conference	1
Skoda Mini[58]	70	Videos	10	ETH Zurich	N/A

TABLE 3.1: Comparison of Existing Datasets

Dataset	**Instances**	**Modality**	**Classes**	**Source**	**Activities per Class**
DAPH NET[60]	237	txt	3	Sourasky Medical Center, Israel	N/A
MHEALTH[65]	120	log	12	University of Granada	N/A
HHAR[66]	4393025	Videos	6	Aarhus University Denmark	1
DSAD[67]	9120	text	8	Bilkent University	1
REAL DISP[68]	1419	text	12	University of Granada	1
tiny virat (UCF)[69]	18752+ 7425+ 11327	Videos	26	University of Central Florida	1

3.2 Local Dataset

3.2.1 Dataset Details

We collect data of 10 different activities from 3 different sources. First, we gather videos from YouTube of the mentioned activities, using approximately 200 videos ranging from 10 seconds to 5 minutes. Second, we will use existing datasets to take required activity samples from each dataset. Third, we record videos for each activity in an indoor setting with consent from the subjects involved as shown in 3.2.1. 50 subjects are recruited for video recording. Video length ranges from 5 seconds to 15 seconds. The mobile camera is of resolution 1280 x 720 and frame rate of 30 fps.

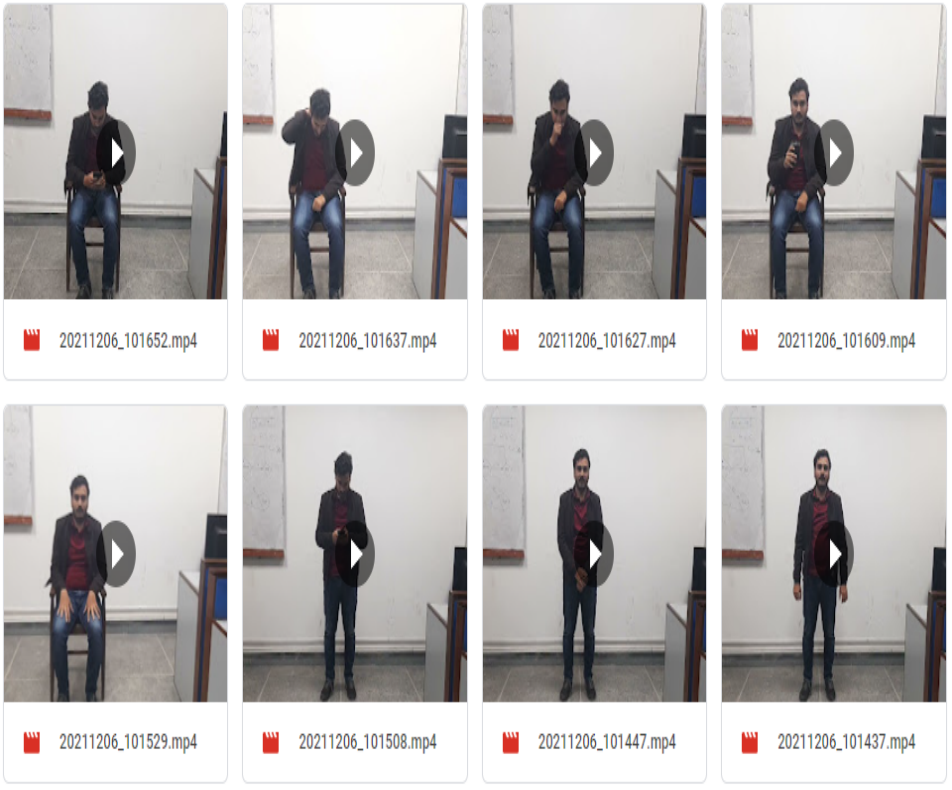


FIGURE 3.1: Videos in Personal Dataset

3.2.2 Data Annotation

We will eliminate videos that are not connected to our chosen activity. With the knowledge that every video must have at least one label, all videos will be played and clipped at the fixed temporal length. However, finding YouTube videos only containing scenes related to our activity class is difficult. As a result, we'll manually cut the video segments from entire videos within those temporal boundaries

where the activity is performed. As a result, each clip is linked to a ground truth activity label.

3.2.3 Naming Convention

Our dataset is divided into 18 folders, each containing clips from a different activity class. Each clip's name is formatted as follows: X_Y.mp4, where X is the name of the folder and Y is the clip number. The clips are all in the.mp4 format.

Chapter 4

Proposed Methodology

The camera is set up in the environment for visual input. The video stream is sent to the application. On this video, model is run for activity recognition and the result is shown on the screen as well as saved in the database. The logs in the database are further used for analysing and detecting any anomaly. Sleep pattern is also analysed and forecasting is done. This methodology is shown in Figure.

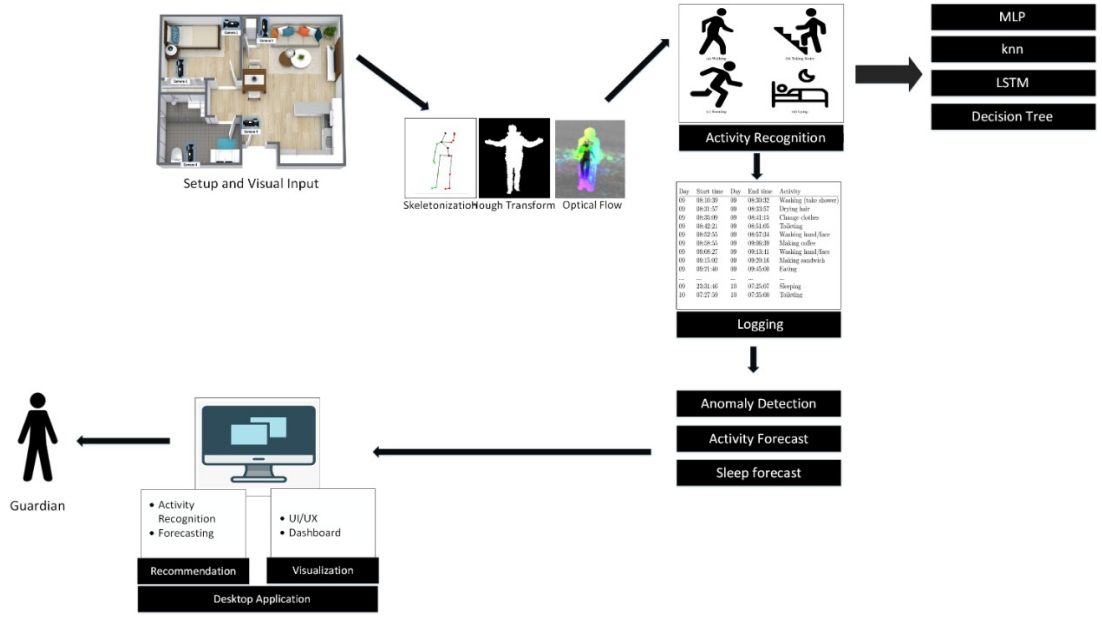


FIGURE 4.1: General Flow Diagram

4.1 Activity Detection using Optical Flow

4.1.1 Dataset

For activity detection, the KTH dataset is used. It includes six activities: walking, jogging, running, boxing, hand-waves, and hand clapping. Each task is carried out by 25 different people, and the environment is deliberately changed for each actor. There are four different scenarios: outside, outdoor with scale, outdoor with scale, and outdoor with scale. Outdoors with various outfits and indoors. The total number of videos available is as a result: 25x4x6 = 600. The videos have a 25fps frame rate and a resolution is 160x120.

4.1.2 Dataset Division

The dataset is divided into folders by following the given structure.

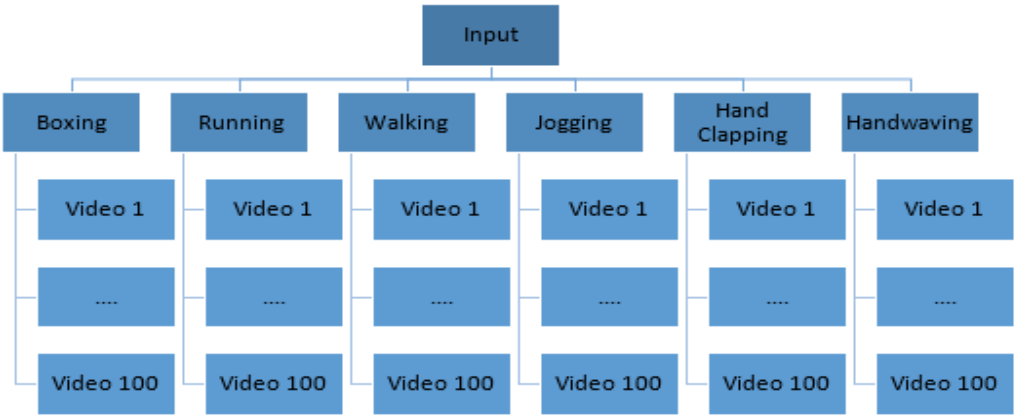


FIGURE 4.2: Folder Structure for Dataset

Further, the frames are extracted from each video and put into their respective folders automatically.

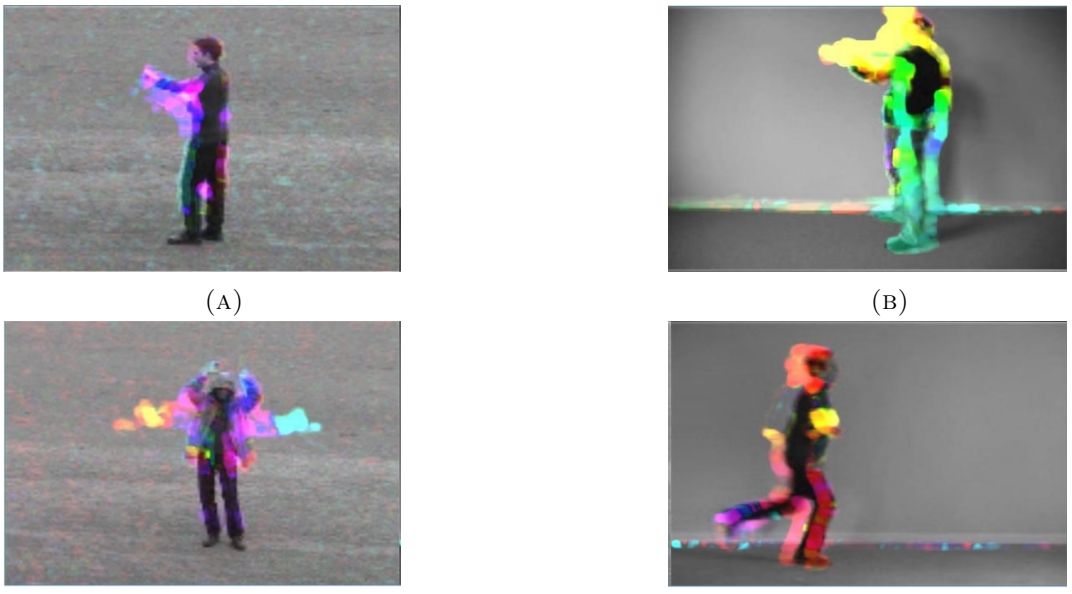
4.1.3 Optical Flow

Optical flow is the movement of objects between frames in a series induced by the relative movement of the item and the camera[70]. Optical flow works on several assumptions:

- Between consecutive frames, the pixel intensities of an entity do not change.
- The mobility of neighbouring pixels is similar.

We employed the dense Farneback approach[71]. In dense optical flow, all points are examined, and then pixel intensity differences between the two frames are

identified, resulting in an image with highlighted pixels, which is subsequently converted to HSV format for easy visualization[72].



(A) (B)

(C) (D)

FIGURE 4.3: Results Obtained from Optical Flow

It uses an array of flow vectors to calculate the magnitude and direction of optical flow vectors, i.e., (dx/dt, dy/dt).The angle (direction) of flow is visualised by hue, and the distance (magnitude) of flow is visualised by the value of HSV colour representation. The strength of HSV is adjusted to 255 for optimal visibility.
OpenCV has a function *cv2.calcOpticalFlowFarneback* for investigating dense optical flow in a greater depth. The result of the Optical flow is shown in Figure 4.3.
In our project, each frame is resized to the size of 600 x 450 and optical flow is applied on each 8th frame. The difference between the magnitude and angle of two frames is calculated and stored in a csv file. Furthermore, each frame is divided into 9 parts as shown in Figure 5 and then each part's angle and magnitude are calculated as its features.

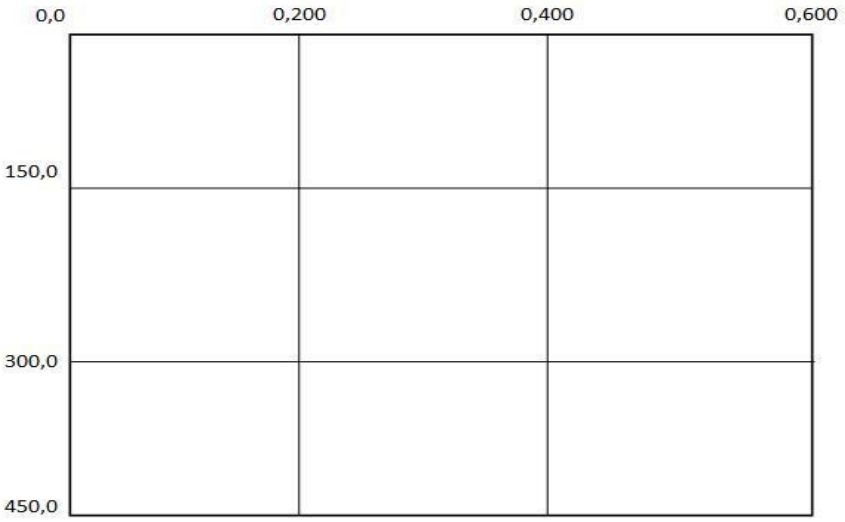


FIGURE 4.4: Frame Dimension

4.1.4 Multi-Layer Perceptron (MLP) Model

MLP uses perceptrons for classification purposes[73]. A perceptron is a linear classifier, which means it is an algorithm that classifies data by drawing a straight line between two categories. A feature vector x is typically multiplied by weights w and then added to a bias b as input[74]. A multilayer perceptron (MLP) is a deep, artificial neural network that is composed of more than one perceptron. They are composed of an input layer to receive the signal, an output layer that makes a decision or prediction about the input, and in between those two, a variable number of hidden layers.Any continuous function can be approximated using MLPs with one hidden layer. The signal flow proceeds from the input layer

via the hidden layers to the output layer in the forward pass, and the output layer's decision is compared to the ground truth labels.

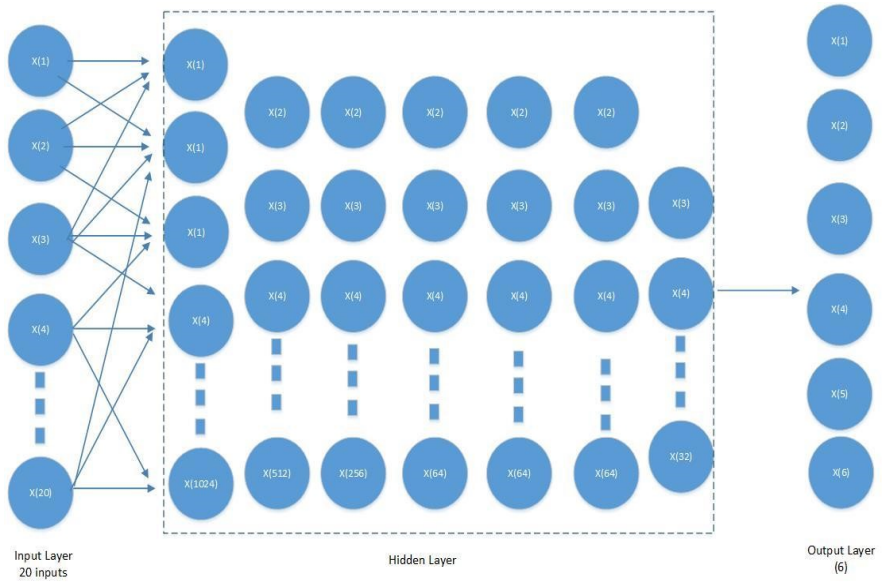


FIGURE 4.5: Typical Architecture of the MLP Model

In the backward pass, partial derivatives of the error function with respect to the various weights and biases are back-propagated through the MLP using backpropagation and the chain rule of calculus is used[75]. The operation of differentiation produces a gradient, or error landscape, along which the parameters can be modified as the MLP approaches the error minimum. This can be done with any gradient-based optimization algorithm such as stochastic gradient descent.
Dataset was divided into test and training sets using 20 and 80 ratio[76]. We used MLP with 20 inputs in the input layer and hidden layers as shown in Figure 4.5. Output layers have 6 outputs using rectified linear unit (ReLu) activation function whose equation is shown in equation 4.1.

$$y = max(0, x) \tag{4.1}$$

It can be seen visually in Figure 4.6

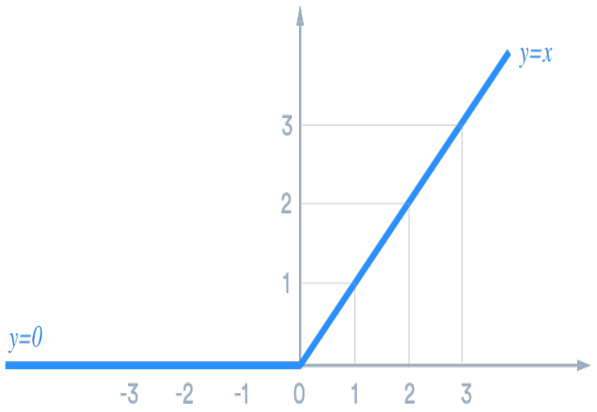


FIGURE 4.6: Rectified Linear Unit (ReLu) Activation Function

4.1.5 Results

Precision (P) or detection rate defines the percentage of accurately classified instances to the total labelled instances. It signifies the true positive value and can be used to measure the prediction's model. It is illustrated below:

$$Precision(P) = TP/(FP + TP) \tag{4.2}$$

Recall (R) or Sensitivity defines as the proportion of labelled occurrences to all instances. The R measure, which is commonly defined as the true positive figure, denotes the predictions' model. which is defined by:

$$Recall(R) = TP/(FN + TP) \tag{4.3}$$

The confusion matrix shows the results of each class. 74% accuracy is achieved by the model as shown in Figure 8 below:

TABLE 4.1: Results Obtained from the MLP Model

	Precision	**Recall**	**F1-score**	**Support**
Boxing	0.92	0.93	0.92	1095
Handclapping	0.83	0.86	0.85	1054
Handwaving	0.88	0.89	0.89	1379
Jogging	0.53	0.47	0.50	1093
Running	0.54	0.51	0.52	927
Walking	0.68	0.73	0.70	1644
Accuracy			0.74	7192
Macro Avg	0.73	0.73	0.73	7192
Weighted Avg	0.74	0.74	0.74	7192

4.2 Gait Based Activity Recognition

Every person has a unique gait means their walking style is different and based on the gait features extracted gait analysis can be used to find out what kind of activities are being performed by the person. This method applies these concepts to extract features from videos to detect activities[77]. This works in the following steps: first, videos are converted into frames and saved in folders then, the background is removed from those frames to obtain silhouette images. After that skeletonization is applied, then Hough transform is used to extract features. In the end, those features are stored in a file that can be trained to detect activities.

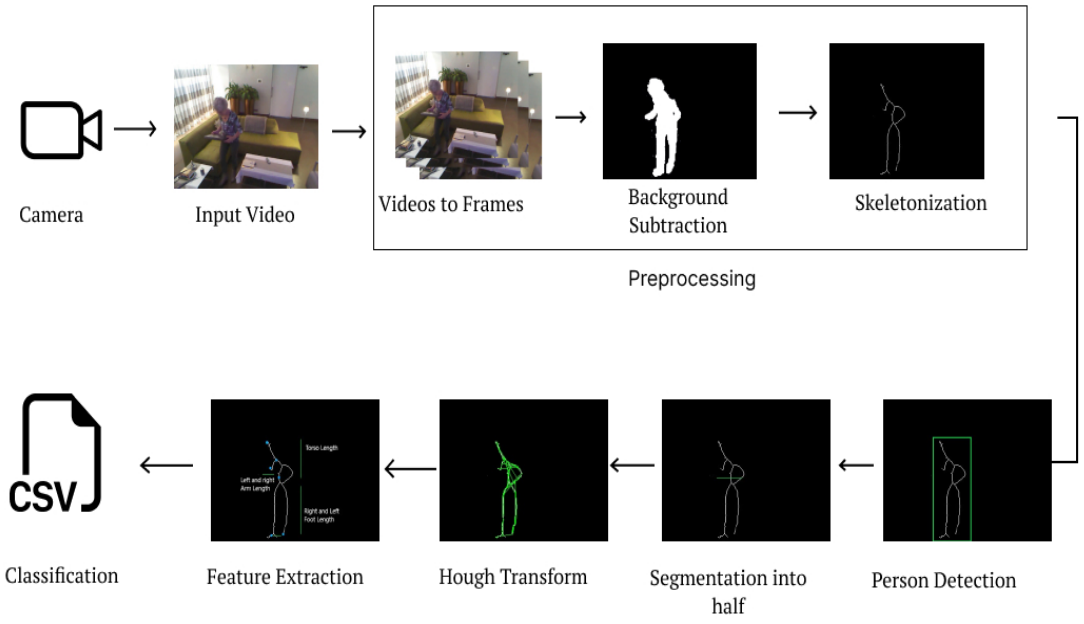


FIGURE 4.7: methodology of Gait Based Activity Detection

4.2.1 Background Subtraction

Background subtraction is used in detecting moving objects in a video. By removing the background from an image all of the unnecessary objects are removed except t the moving object that is being detected. To remove the noise for better background subtraction median filter is used. This filter runs through each pixel and then arranges

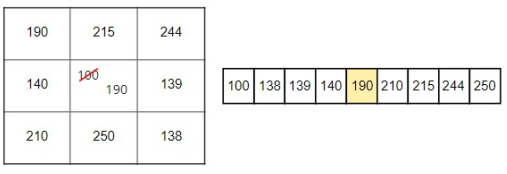


FIGURE 4.8: Applying Median Filter

all neighboring pixels in a numerical order to find the median values that is replaced with the current pixel value. It removes salt noise and is better in preserving edges.

Next to get silhouette thresholding is used. It based on pixel intensities will binarize the image[78]. So if intensity of image pixel is greater than threshold it mark it as white and black otherwise[79]. Here we use Otsu's thresholding approach as it calculates threshold automatically.

FIGURE 4.9: Background Subtraction

4.2.2 Skeletonization

To extract features of object it is first converted into a skeleton using medial axis morphology[80]. It is the set of all points that have more than one closest point on the object's boundary[81]. It is a 1-pixel wide skeleton of the object and the connectivity is the same as the original[82].

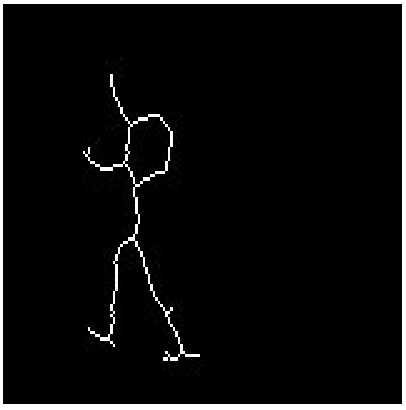


FIGURE 4.10: Skeleton using Medial Axis Morphology

4.2.3 Dividing Human Skeleton

At this stage, the skeleton is divided into two parts by calculating the centroid found from the center of the mass. The area above the centroid is assumed an upper body that contains: the head, neck, torso, and arms while the area below the centroid is considered a lower body that has: legs and feet.

FIGURE 4.11: Skeleton Divison

4.2.4 Hough Transform and Feature Extraction

In this step using Hough transform straight lines are found. Hough transform will give the straight lines on the skeleton and from that, we can select the largest lines to find the desired angle features like the angle between torso and left arm[83], the angle between torso and right arm and angle between two feet. We can find the angle by using the formula:

$$90 + \theta = \Theta \tag{4.4}$$

(A) (B)

FIGURE 4.12: Lines Detected using Hough Transform

Lines given by Hough transform are divided into upper body lines or lower body lines and all other features are extracted by finding the key points in these lines. A total of 18 features are extracted that include: headPoint_X, headPoint_Y, leftArmPoint_X, leftArmPoint_Y, armJointPoint_X, armJointPoint_Y, centoidPoint_X, centoidPoint_Y, rightFootPoint_X, rightFootPoint_Y, leftFootPoint_X, leftFootPoint_Y, angleBetweenTorsoAndLeftArm, angleBetweenTwolegs, lengthOfLeftArm, lengthOfLeftLeg, lengthOfRightLeg and lengthOfTorsoLine. These features are stored in a csv file[84].

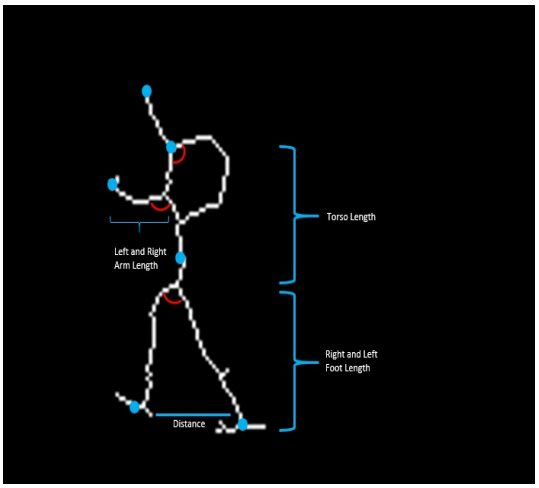


FIGURE 4.13: Gait Features Extraction

4.2.5 Classification

After all the features are stored in CSV file different classification techniques are used to find patterns among those features so that the same patterns can be detected in the future data. These classification algorithms categorize our input data into 6 classes so classification techniques that support multi-class are used. Different classification techniques are used to build these classifiers[85].

- KNN stands for K-nearest neighbour takes a tuple from the dataset makes a point in the n-dimensional space. Whenever a new unknown tuple comes, KNN method searches the space, to get the k nearest instance. The new instance is assigned the label that comes mostly in the nearest neighbours. In order to find the nearest neighbours, Euclidean distance can be used to find the nearest neighbours[86].

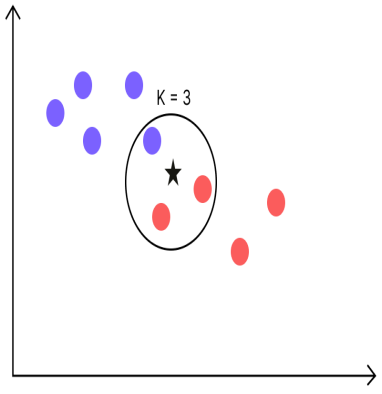


FIGURE 4.14: Example of K-Nearest Neighbour

- The decision tree is a tree-like structure[87]. It is a type of inductive reasoning algorithm that is similar to a flow chart. The class of a tuple is determined by searching the tree from root till the leaf node. The leaf nodes show the class type of the tuple while there is a classification rule at each other node that represents a particular attribute and it splits the data according to that particular attribute value[88]. The classification rule is the conjunction of attributes so that the tree can also be represented as a disjunction expression. In order to split at each level information, the gain is used as an attribute selection standard. For testing the tuple value is again tested from root to leaf, according to the tested tuple attribute value[89].

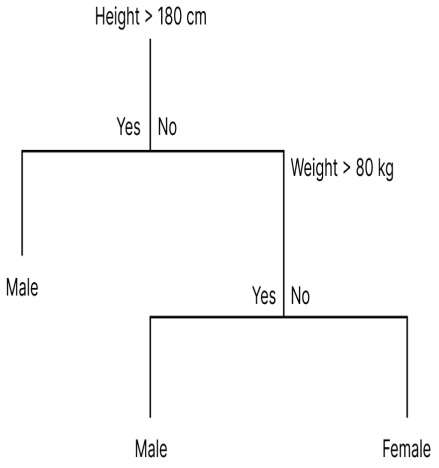


FIGURE 4.15: Example of Decision tree

- Naive Bayes which is a classification method is based on Bayes' Theorem. It is a statistical classification method. In Naive Bayes, there is an assumption that each particular feature of a class is unrelated to the other features. It uses the techniques of statistics and probability to classify the unknown data. Theoretically, the independence of features represents that the error rate of the Naive Bayes classifier is the smallest. The Bayes' Theorem calculates the chance of an event occurring given the probability of a previous event. The mathematical form of Bayes' Theorem is as follows:

$$P(L|M) = \frac{\mathrm{P(M|L) + P(L)}}{\mathrm{P(M)}} \tag{4.5}$$

4.2.6 Dataset

The dataset used for activity detection is KTH which is a standard dataset for activity detection and it has 6 action classes: hand clap, hand-wave, jog, walk, box and run as shown in Figure 4.16. There is a total of 25[90] subjects that perform these actions in 4 different setting variations: indoor, outdoor, outdoor with different clothes, and outdoor with scale variations. Total videos in this case will be 25x4x4 = 600. The video frames have a resolution of 160x120 with a frame rate of 25fps. After extraction of features from frames there are a total of 171,688 training samples and 42,922 testing samples[91].

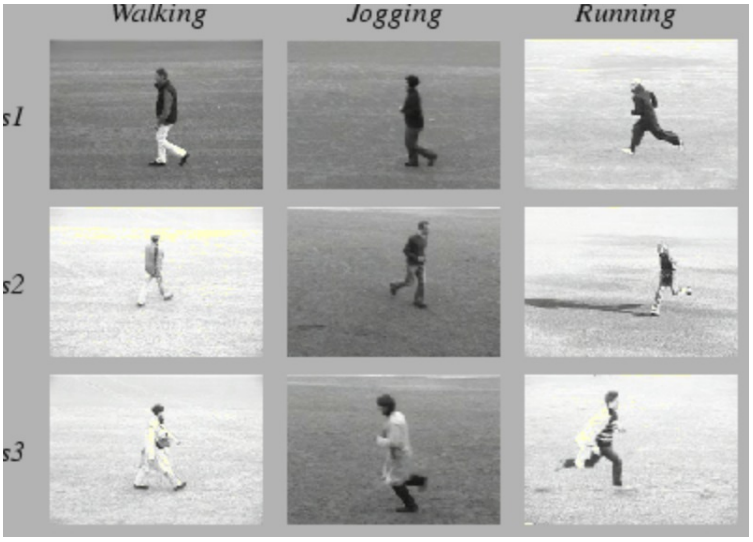


FIGURE 4.16: KTH Dataset

Dataset is divided into train and test folders shown in Figure 4.16.

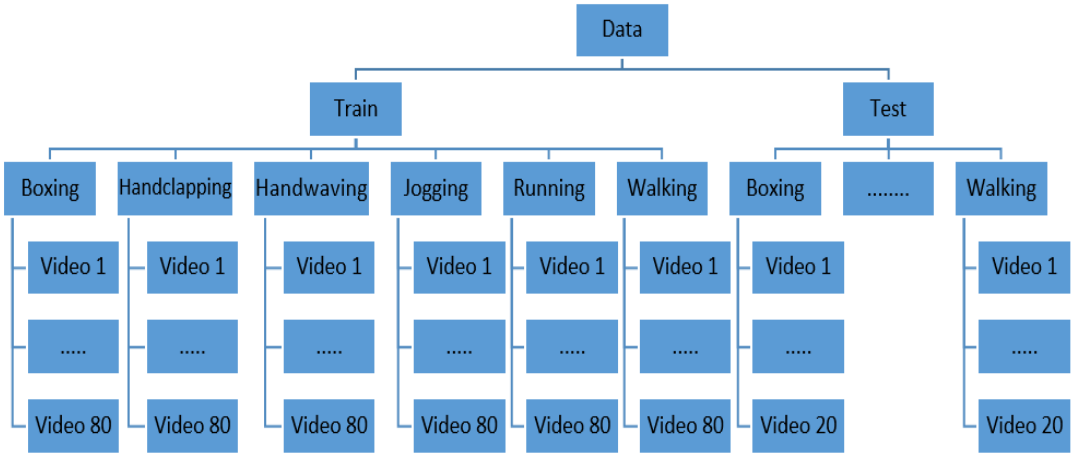


FIGURE 4.17: Folder Structure for Dataset

Further, the frames of each video are extracted and stored in respective folder to extract features.

TABLE 4.2: Training and Testing samples of each class

Class	**Train**	**Test**
Boxing	33,205	8,301
Handclapping	40,677	8,135
Handwaing	41,257	10,314
Jogging	20,973	5,243
Running	20,727	4,145
Walking	33,913	6,782

4.2.7 Results

For activity detection some famous classification techniques such as KNN, Decision Tree and Naive Bayes have been used. The results show that 56.7% accuracy is achieved by using our features with the KNN model. An accuracy of 0.29% is achieved using the Naive Bayes while Decision Tree gives an accuracy of 0.34%. To compare the result, different evaluation matrices are used including Accuracy, f1-score, recall, precision and false positives. The result is evaluated using the following matrices: Accuracy is the measure of correct classification of videos by the proposed system and it is defined as follows:

$$Accuracy = \frac{\text{TP} + \text{TN}}{\text{Total Sample in the dataset}}$$

Precision is the proportion of positive activity that have actually the said activity. It is calculated using the following:

$$Precision = \frac{\text{Correctly classified positive images}}{\text{Total classified positive images}}$$

F1-score is the harmonic mean of precision and recall and it is defined as

$$F_1 = 2\frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

Recall is the ability of proposed model to detect all the potential activities. It is calculated as follows:

$$Recall = \frac{\text{Correctly classified positive images}}{\text{Total positive images in the dataset}}$$

Support is the number of occurrences of each class in y_true and it can be calculated as follows:

$$Support = \text{occurrences of each class (y\_true)}$$

Sensitivity is the number of positives returned by the model as it is the ability of model to find all positives. It is calculated as follows:

$$Sensitivity = \frac{\text{TP}}{\text{TP + FN}}$$

Specificity is the number of negatives returned by the model as it is the ability of model to find all negatives. It is calculated as follows:

$$Specificity = \frac{\text{TN}}{\text{TN + FP}}$$

AUC (Area Under ROC curve) is based on threshold values. It is used to see how good model is in distinguishing different classes.

Table 4.3: Comparison of different classification algorithms

	KNN	**Naive Bayes**	**Decision Tree**
Accuracy	0.57	0.29	0.34
Precision	0.57	0.27	0.21
Recall	0.57	0.29	0.35
F1-score	0.57	0.28	0.26
Support	42922	42922	42922
Sensitivity	0.53	0.26	0.28
Specificity	0.91	0.85	0.86
AUC	0.72	0.55	0.57

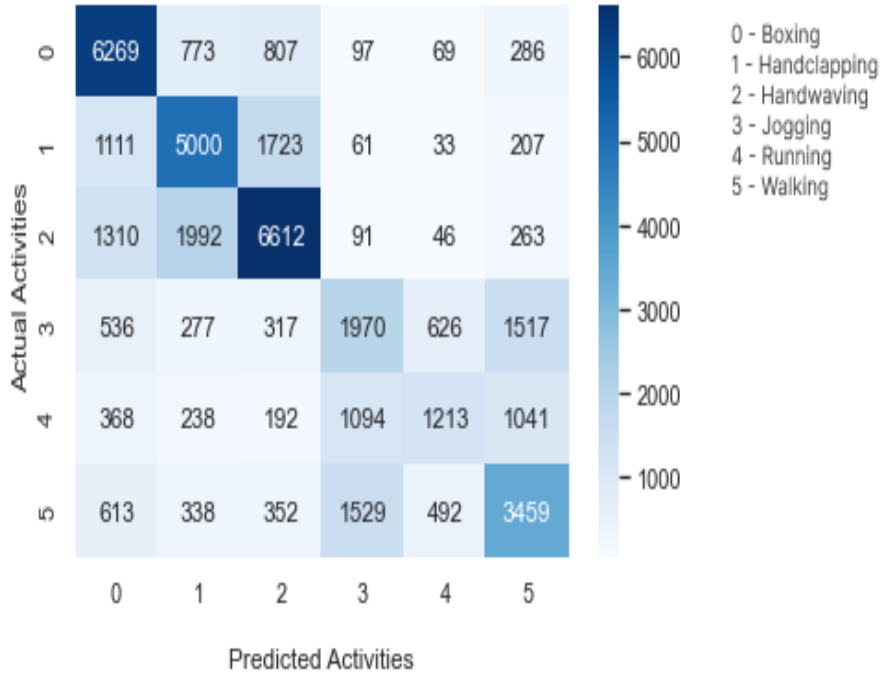


FIGURE 4.18: Confusion Matrix of KNN

4.3 Activity Recognition using RNN

4.3.1 Dataset

Our dataset consists of videos with mp4 extension and it consists of 10 classes namely walking, eating, laying, sitting, leaving, taking pills, using telephone and drinking. The data is taken from Toyota Smart Dataset which consists of daily activities of 18 subjects. The age of subjects is in the range of 60-80. For data cleaning, only 10 relevant activities are selected. To balance data, all classes are given the same number of videos by selecting the minimum possible videos number. The training folder has 11,000 videos with approximately 1100 videos in each folder. The test folder has 1200 videos with approximately 120 videos in each folder.

The dataset is divided into train and test folders with further division into the labelled folders for each given class as shown in Figure 4.20.

Train

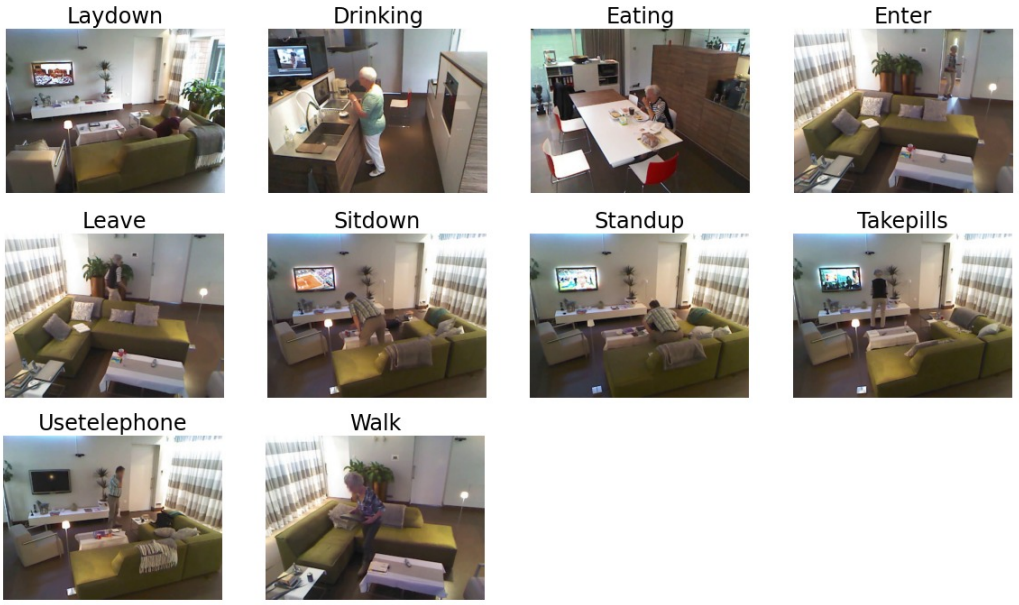


FIGURE 4.19: Training dataset structure

Test

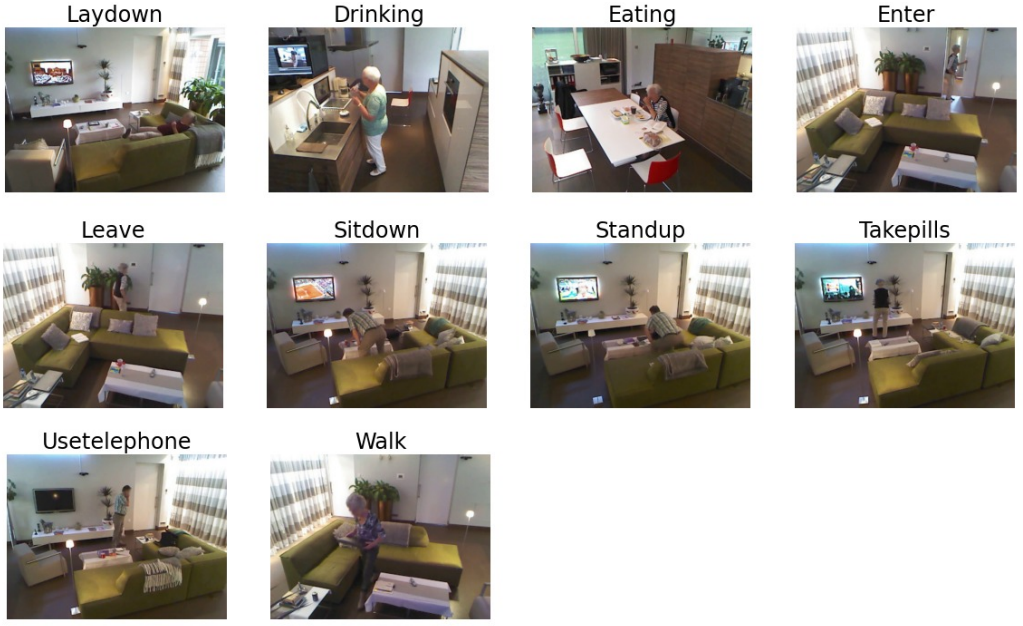


FIGURE 4.20: Testing dataset structure

4.3.2 Frames Generation

We extracted frames from each video and stored the results in train and test folders. The videos are sub sampled to 40 frames per video during training phase. Frames have original of dimensions 640 x 480. The size of reduced to 320 x 240 before saving in order to save space. Train and test are divided on ratio of 1 to 3 randomly. Random text files are generated using python and the text files are used to separate videos for extracting frames using FFmpeg. Following are a few frames of class Laydown.

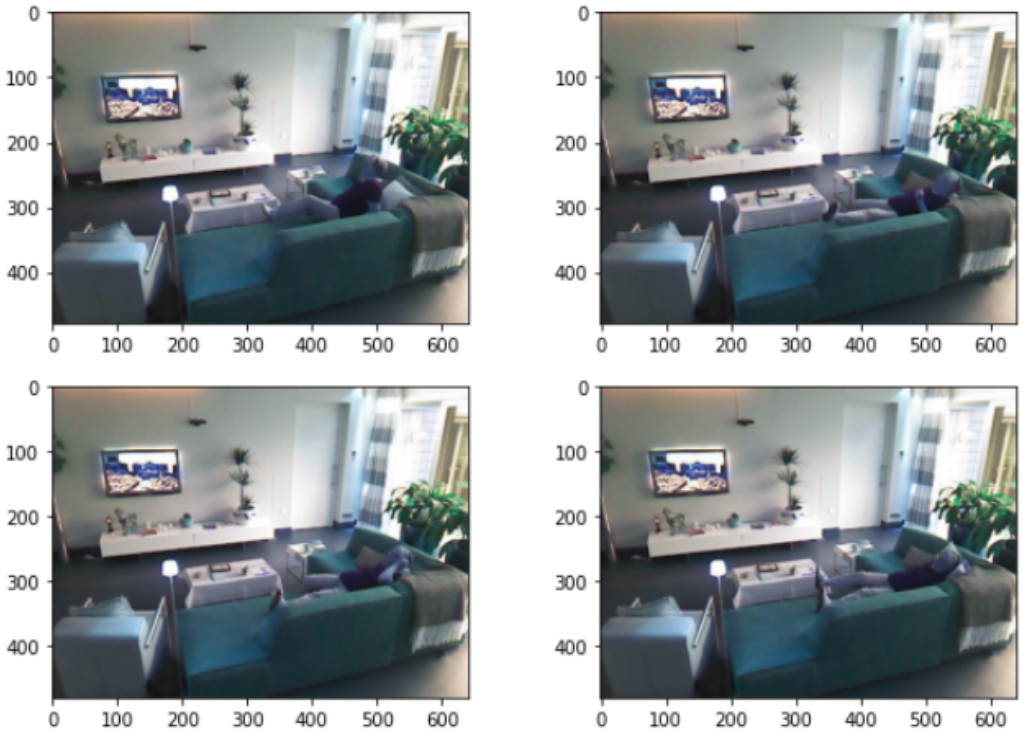


FIGURE 4.21: Frames of class Laydown

4.3.3 RNN Architecture

A Recurrent neural network(RNN) uses a looped architecture in order to use past knowledge as continuing information. In this case, RNN is used for sequential data that by using past images in memory extracts the temporal correlation between them[92].

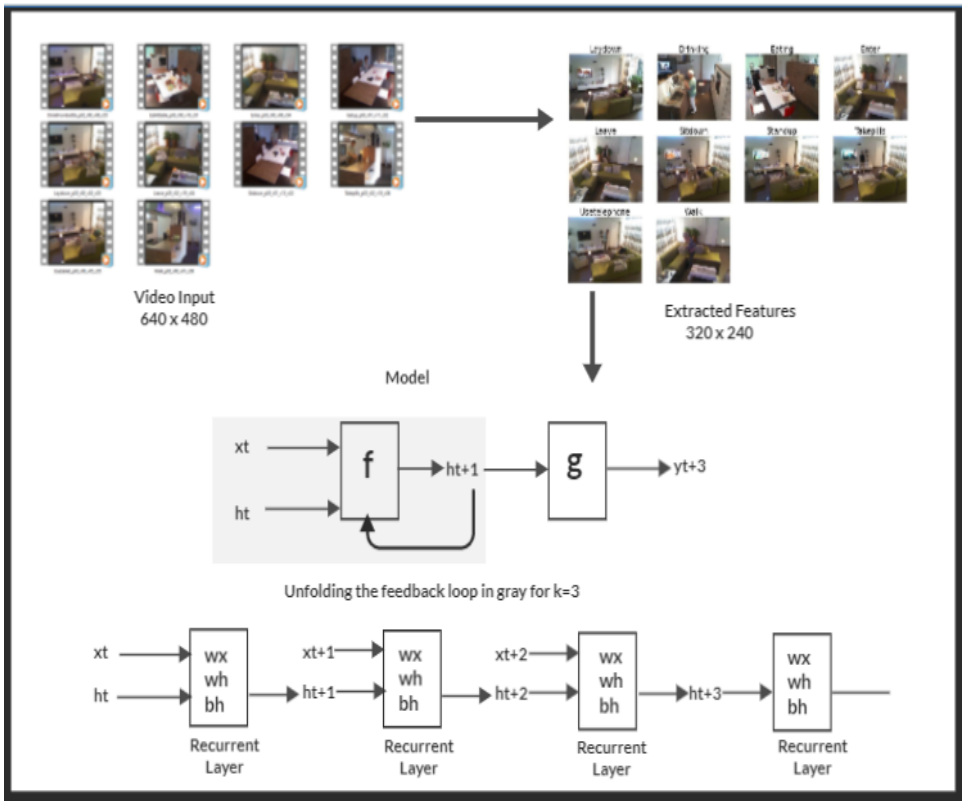


FIGURE 4.22: RNN Architecture Diagram

Usually, the feed-forward neural networks work for independent data points. Still, if there is any dependency between data points that one point depends upon a previous data point then a traditional feed-forward neural network does not work. The idea of 'memory' is used in RNN to remember the information from the past data points and give it input to the next state.
As seen in the Figure 4.22, a simple RNN has a feedback loop. The grey rectangle represents a feedback loop that can be unrolled in three-time steps to form the second network seen above. The architecture can be changed so that the network unfolds in k time steps. The following notation is used in the diagram:

- The input at time step t is x_t. We'll assume x_t is a scalar field with a single feature to keep things easy. This concept can be extended to a n-dimensional feature vector.
- The network's output at time step t is y_t. Although the network can have several outputs, the assumption is single output in this case.
- The hidden units/states' values are stored in the h_t vector at time t. This is also known as the current context. The number of hidden units is m. The h_0 vector is set to zero.

- The weights correlated with inputs in recurrent layer are w_x
- The weights in recurrent layer that are correlated with hidden units are w_h
- The weights correlated with hidden to output units are w_y
- The bias correlated with the recurrent layer is b_h
- The bias correlated with the feedforward layer is b_y

The output at time step n+1 can be obtained by unrolling the network for n time steps at each time step. The unrolled network resembles the feedforward network. An operation taking place in the unrolled network is indicated by the rectangle. For instance, consider the activation function f:

$$h_t + 1 = f(h_t, , w_x, w_h, b_h) = f(w_h h_t + w_x x_t + b_h) \tag{4.6}$$

The output y computed at time t is as:

$$y_t = f(w_y, h_t) = f(w_y.h_t + b_y) \tag{4.7}$$

In RNN during a feedforward pass the values of hidden units and the output after n time steps are computed by the network. All the weights that are correlated with the network are temporally shared. There are two sets of weights in each recurrent layer; one is for the hidden unit and another one for the input unit. The last layer that gives the final output for the nth step looks like a simple layer from a feedforward network.

In an artificial neural network, the backpropagation algorithm is modified so that there is an element of unrolling in time to train the weights. This algorithm is known as backpropagation in time of the BPTT algorithm as it computes the gradient vector. For training, the network pseudo-code is given below. Users can select the value of n for training.

1. Keep on repeating until the stopping condition is met:
2. All h are set to 0.
3. Use loop for t = 0 to m-n.
4. To compute all h and y, forward propagate the network over the unrolling network for n time steps.
5. Find the error as: e=y_t+k-p_t+k
6. Update the weights by backpropagating the error across unrolling network.

4.3.4 Results

The results showed that training accuracy of **0.50%** is achieved for the model used. To compare the result, different evaluation matrices are used including Accuracy, f1-score, recall, precision and false positives. The result is evaluated using the following matrices:

Accuracy is the measure of correct classification of videos by the proposed system and it is defined as follows:

$$Accuracy = \frac{\text{TP} + \text{TN}}{\text{Total Sample in the dataset}}$$

Precision is the proportion of positive activity that have actually the said activity. It is calculated using the following:

$$Precision = \frac{\text{Correctly classified positive images}}{\text{Total classified positive images}}$$

F1-score is the harmonic mean of precision and recall and it is defined as

$$F_1 = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

Recall is the ability of proposed model to detect all the potential activities. It is calculated as follows:

$$Recall = \frac{\text{Correctly classified positive images}}{\text{Total positive images in the dataset}}$$

The results based on these metrics are shown in Table 4.4.

Table 4.4: Results Of RNN

	Precision	**Recall**	**F1-score**
Drinking	0.22	0.53	0.31
Eating	0.37	0.65	0.47
Enter	0.34	0.40	0.37
Leave	0.57	0.45	0.50
Lay	0.39	0.75	0.51
Sit	0.25	0.02	0.03
Stand	0.00	0.00	0.00
Take pills	0.33	0.09	0.14

Table 4.4: Results

	Precision	**Recall**	**F1-score**
Use Phone	0.00	0.00	0.00
Walk	0.25	0.06	0.09

After training the training accuracy of **0.50%** as shown in Figure 4.24 and Figure 4.25, achieved which is the accuracy and loss obtained when the model is tested on identical images which were used in training. And testing accuracy of **0.34%** achieved after testing the model on independent images that were not used in training.

Confusion matrix for the result is shown in Figure 4.23 with the mapping used: ['Drink':0, 'Eat':1, 'Enter':2, 'Leave':3, 'Lay':4, 'Sit':5, 'Stand':6, 'pills':7, 'phone':8, 'Walk':9]

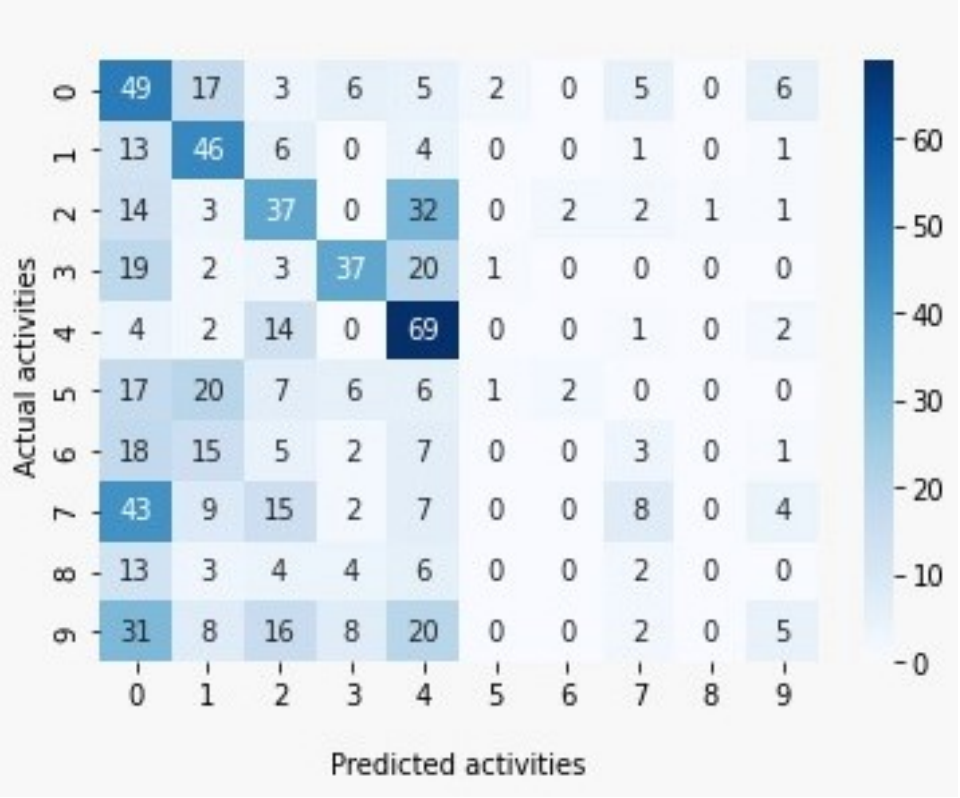


FIGURE 4.23: Confusion Matrix

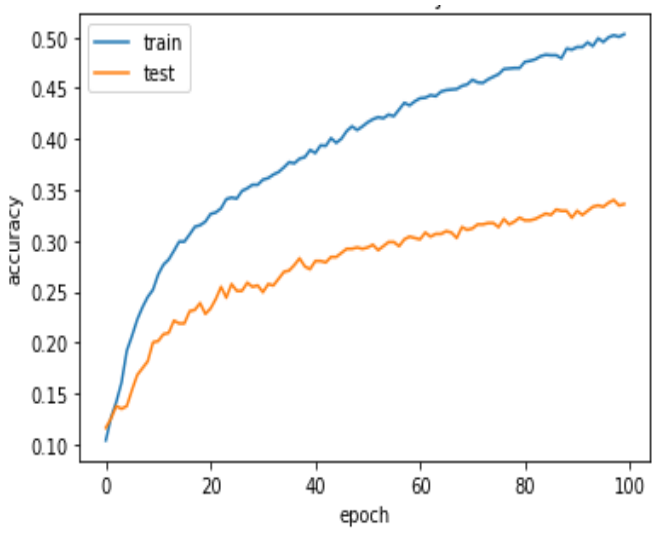


FIGURE 4.24: Training Accuracy

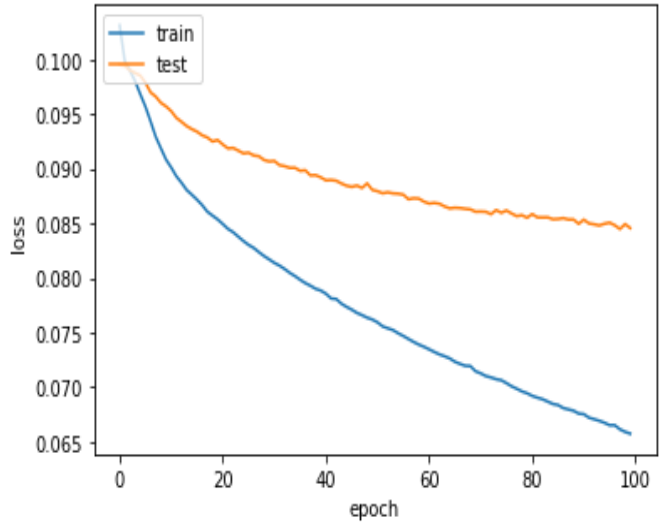


FIGURE 4.25: Training Loss

4.4 Activity Recognition using LSTM

4.4.1 Dataset

The dataset consists of assembled videos with the mp4 extension. The videos are taken from Toyota Smart Dataset, UR Fall Dataset and our Personal Dataset.It consists of 10 classes namely walking, eating, laying, sitting, leaving, taking pills, falling and drinking. The data that is taken from Toyota Smart Dataset consists of daily activities of 18 subjects. The age of subjects is in the range of 60-80. For data cleaning, only 10 relevant activities are selected. To balance data, all classes are given same number of videos by selecting minimum possible videos number. The data from Personal Dataset includes data of 10 subjects. 5 activities are taken from this dataset namely, drinking, eating, walking, standing and sitting. Video

resolution is 1280 x 780 and the frame rate is 30 fps. Sub sampling is done and 40 frames are extracted from each video.
Finally, videos from the UR Fall dataset are used. This dataset has 70 sequences (30 falls + 40 everyday activities). Falling events are captured using two Microsoft Kinect cameras and accelerometric data. Only one device (camera 0) and an accelerometer are used to capture ADL occurrences. PS Move (60Hz) and x-IMU (256Hz) devices were used to collect sensor data. The following is how the data is arranged. Each row provides a succession of depth and RGB pictures for cameras 0 and 1 (positioned parallel to the floor and ceiling, respectively), as well as synchronisation and raw accelerometer data. Each video stream is saved as a png picture sequence in a separate zip package.Only fall films from RGB data, Camera 0, are included in our collected dataset SUR.
The dataset is divided into train and test folders. The training folder has 11,000 videos with approximately 1100 videos in each folder. The test folder has 1200 videos with approximately 120 videos in each folder. The further division into labelled folders for each given class is shown in Figure 4.26 and Figure 4.27.

Train

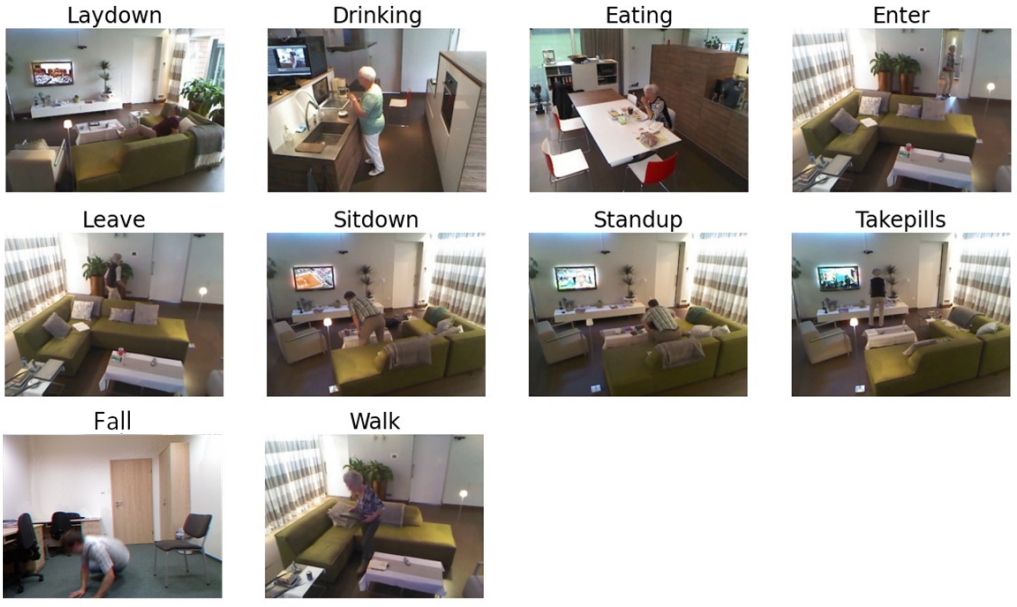


FIGURE 4.26: Training dataset structure

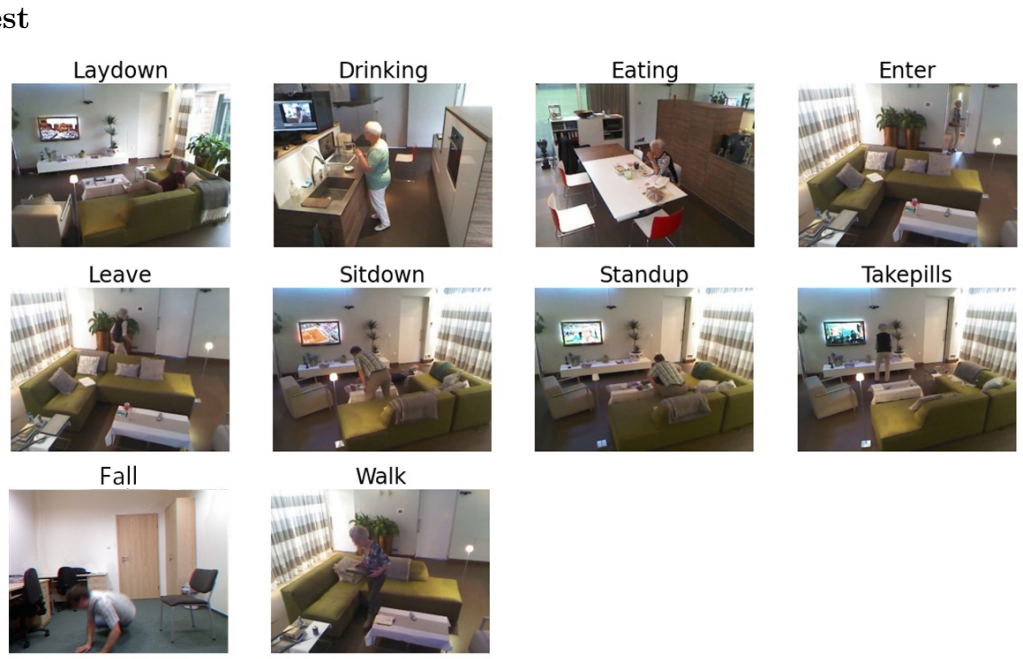


FIGURE 4.27: Testing dataset structure

4.4.2 Dropout for LSTM

Dropout layer had already been shown to be extremely successful in deep convolutional neural networks, but there has been little study done on its application in RNNs. Dropout is applied only to the feedforward connections, not the recurrent connections, in order to retain the capacity of RNNs to model sequences. This is to prevent the units from losing all of their data (due to dropout). It's worth noting that the prior research only looked at dropout in an LSTM neuron's output response.

However, because an LSTM neuron is made up of internal cell and gate units, it's better to create efficient dropout methods by looking at the neuron's output as well as its internal structure. In order to solve this difficulty, LSTM employs an in-depth dropout.

4.4.3 LSTM Architecture

For activity detection, we develop a fully linked LSTM network. Figure 4.29 depicts the suggested network's design, which includes three bidirectional LSTM layers, two feedforward layers, and a dense layer that uses the Adam optimizer to make predictions. We employ Python 3, TensorFlow, and Keras in our technique, as well as ffmpeg for data preparation and video categorization. Each video's frame sequence is extracted and stored using FFmpeg. To detect the sequence, the suggested complete connection architecture allows one to fully use the underlying correlations among the frames. Long Short-Term Memory Network (LSTM) is

a recurrent neural network that avoids the vanishing gradient problem by employing Backpropagation Through Time training. It is capable of learning long-term dependence. A typical LSTM neuron is seen in Figure 4.28, which includes an input gate it, a forget gate ftextsubscriptt, a cell ctextsubscriptt, an output gate otextsubscriptt, and an output response htextsubscriptt. The information flow into and out of the cell is controlled by the input and forget gates[93].

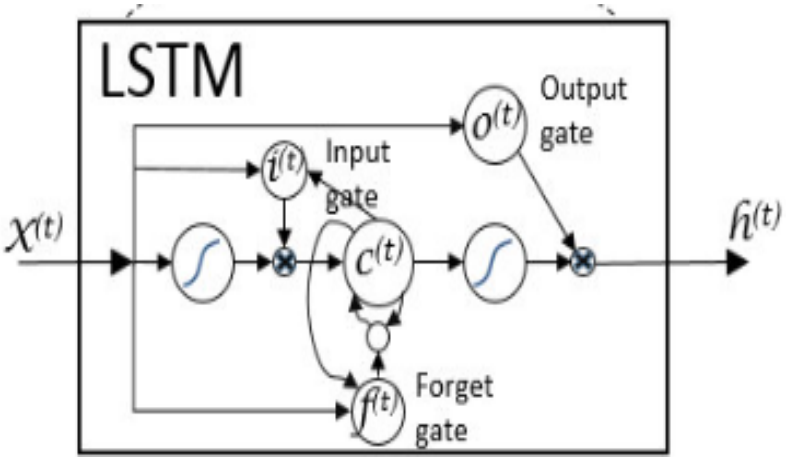


FIGURE 4.28: LSTM Neuron

The output gate determines how much data from the cell is sent to the h_t output. The gradient can traverse through many time steps without fading or bursting because the memory cell has a selfconnected recurrent edge of weight 1. As a result, it overcomes the "vanishing gradient" effect's challenges in training the RNN model.

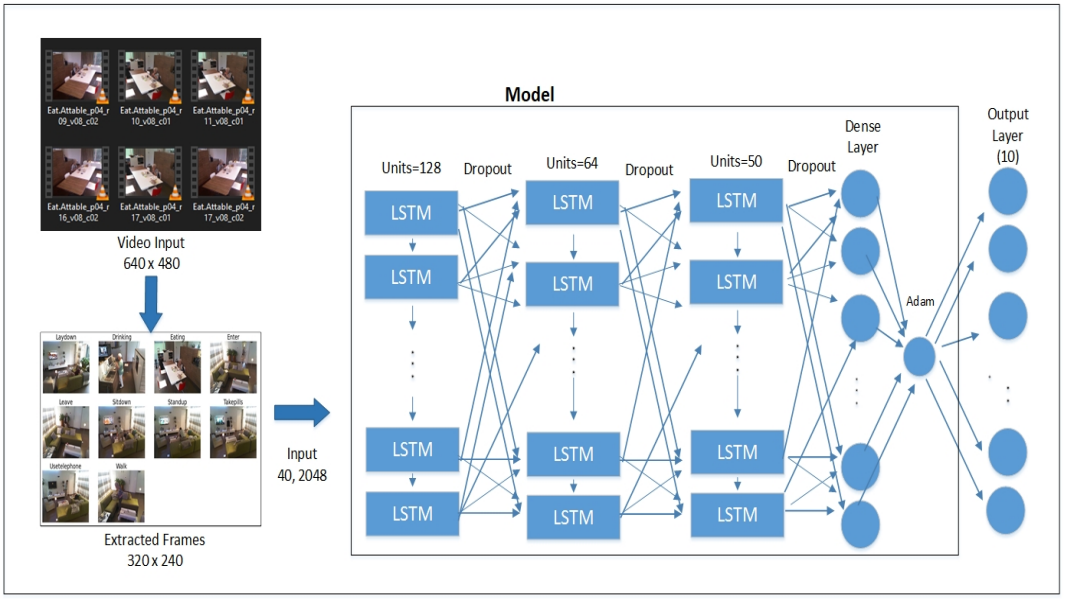


FIGURE 4.29: LSTM Architecture

It may therefore be used to build massive recurrent networks. LSTM networks use memory blocks connected by layers instead of neurons.

A block has gates that control the state and output of the block. Each gate inside a block employs the sigmoid activation units to regulate whether or not they are activated, making the change of state and addition of information flowing through the block conditional[94].
Within a unit, there are three sorts of gates:

- Forget Gate: determines what information to discard from the block on a case-by-case basis.
- Input Gate: determines which values from the input should be used to update the memory state.
- The output gate determines what to output based on the input and the block's memory.

Each unit functions as a little state machine, with weights learnt during the training method.

4.4.4 Results

The results show that accuracy of 38% is achieved for the model used. TDifferent evaluation matrices are used to compare the results, including Accuracy, f1-score, recall, precision, auc, and roc. The following matrices are used to evaluate the outcome: Accuracy is a measure of the suggested system's ability to correctly classify movies, and it is defined as follows:

$$Accuracy = \frac{\text{TP} + \text{TN}}{\text{Total Sample in the dataset}}$$

Precision is the proportion of positive activity that have actually the said activity. It is calculated using the following:

$$Precision = \frac{\text{Correctly classified positive images}}{\text{Total classified positive images}}$$

F1-score is the harmonic mean of precision and recall and it is defined as

$$F_1 = 2\frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

Recall is the ability of proposed model to detect all the potential activities. It is calculated as follows:

$$Recall = \frac{\text{Correctly classified positive images}}{\text{Total positive images in the dataset}}$$

The results based on these metrics are shown in Table 4.5.

TABLE 4.5: Results of LSTM

	Precision	**Recall**	**F1-score**
Drinking	0.18	0.27	0.22
Eating	0.54	0.72	0.61
Enter	0.58	0.64	0.61
Leave	0.50	0.11	0.18
Lay	0.47	0.65	0.54
Sit	0.20	0.27	0.23
Stand	0.36	0.16	0.22
Fall	0.43	0.23	0.30
Take pills	0.00	0.00	0.00
Walk	0.37	0.50	0.42

Confusion matrix for the result is shown in Figure 4.30 with mapping used: ['Drink':0, 'Eat':1, 'Enter':2, 'Leave':3, 'Lay':4, 'Sit':5, 'Stand':6, 'pills':7, 'Fall':8, 'Walk':9]

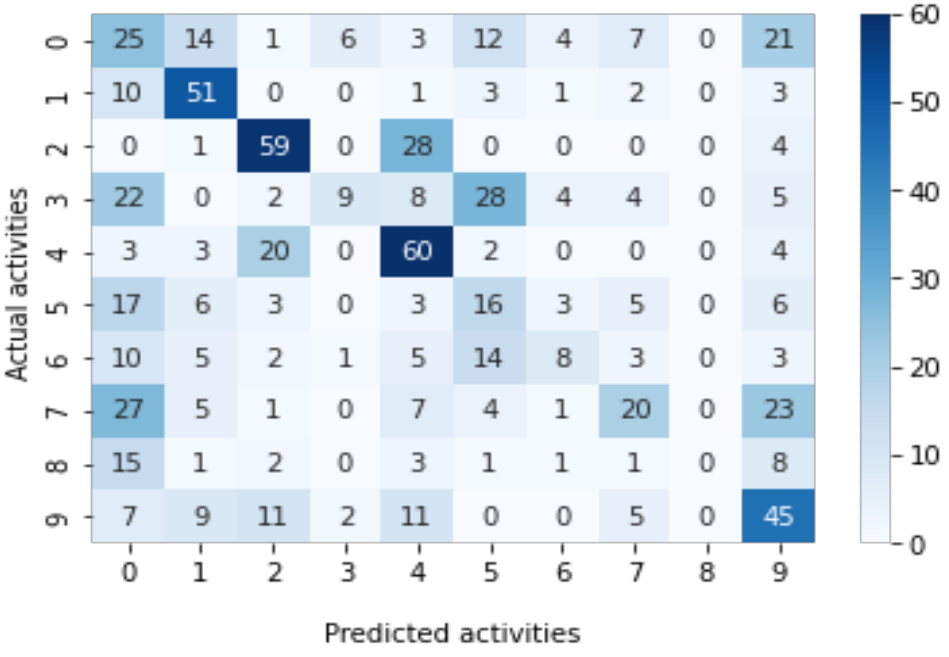


FIGURE 4.30: Confusion Matrix

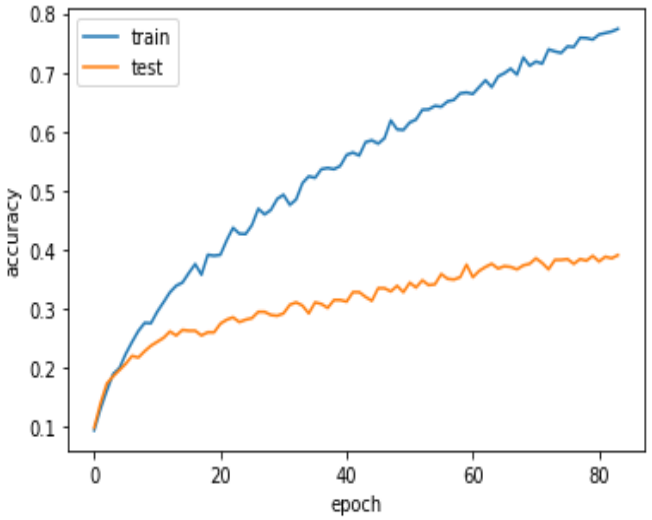


FIGURE 4.31: Training accuracy

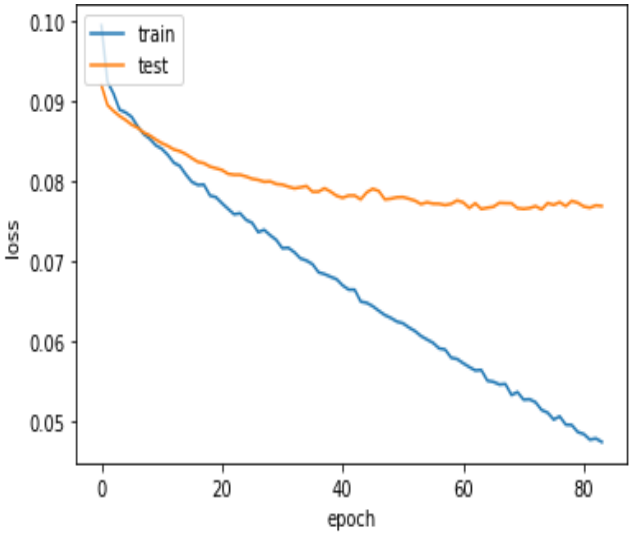


FIGURE 4.32: Training Loss

4.5 Activity Recognition using LSTM-CNN

4.5.1 Dataset

Our dataset consists of videos with mp4 extension and it consists of 10 classes namely walking, eating, laying, sitting, leaving, taking pills, using telephone and drinking.

The dataset is divided into train and test folders with further division into labelled folders for each given class as shown in Figure 4.33 and Figure 4.34.

The data is taken from Toyota Smart Dataset which consists of daily activities of 18 subjects. The age of subjects is in the range of 60-80. For data cleaning, only 10

relevant activities are selected. To balance data, all classes are given same number of videos by selecting minimum possible videos number. The training folder has 11,000 videos with approximately 1100 videos in each folder. The test folder has 1200 videos with approximately 120 videos in each folder.

Train

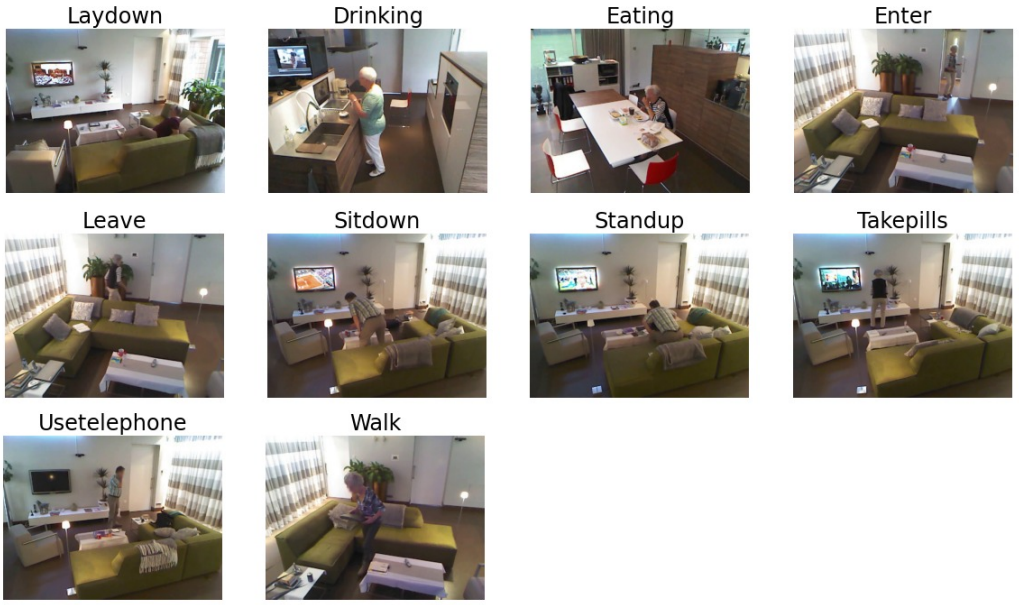


FIGURE 4.33: Training dataset structure

Test

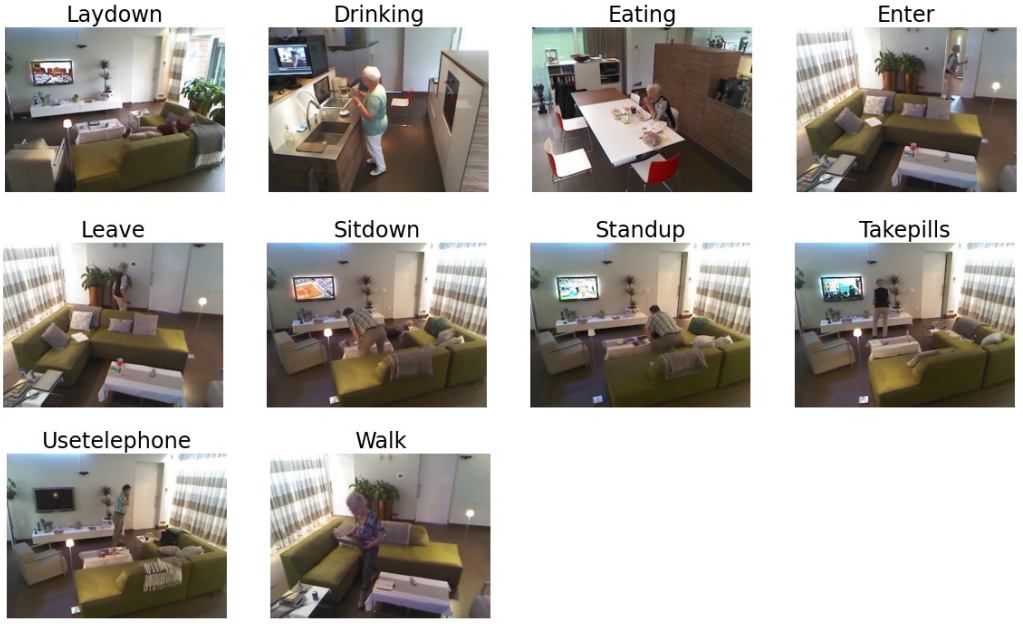


FIGURE 4.34: Testing dataset structure

4.5.2 Extracting Frames

We extracted frames from each video and stored the results in train and test folders. The videos are sub sampled to 40 frames per video during training phase. Train and test are divided on ratio of 1 to 3 randomly. Random text files are generated using python and the text files are used to separate videos for extracting frames using FFmpeg.

4.5.3 Extracting Features

To extract features from the images, a pre-trained model is used provided by Keras which is called Inception-V3[95]. The architecture of Inception V3 model is shown in Figure 4.35. It is a 48 layer deep model provided by Keras whose weights are pre-trained on ImageNet dataset, which is a huge dataset consisting of large number of images of different classes. This saves time time as it provides an already existing architecture which takes less efforts and produce better results.
The frames from one video sequence are loaded and converted into an array using Numpy. The input is then prepossessed according to the model requirement using preprocess_input method from keras. The model gives prediction for the input which is used as features sequence and stored as features in sequence file in a standard binary file format.We use the result of the feature extraction in next stage.

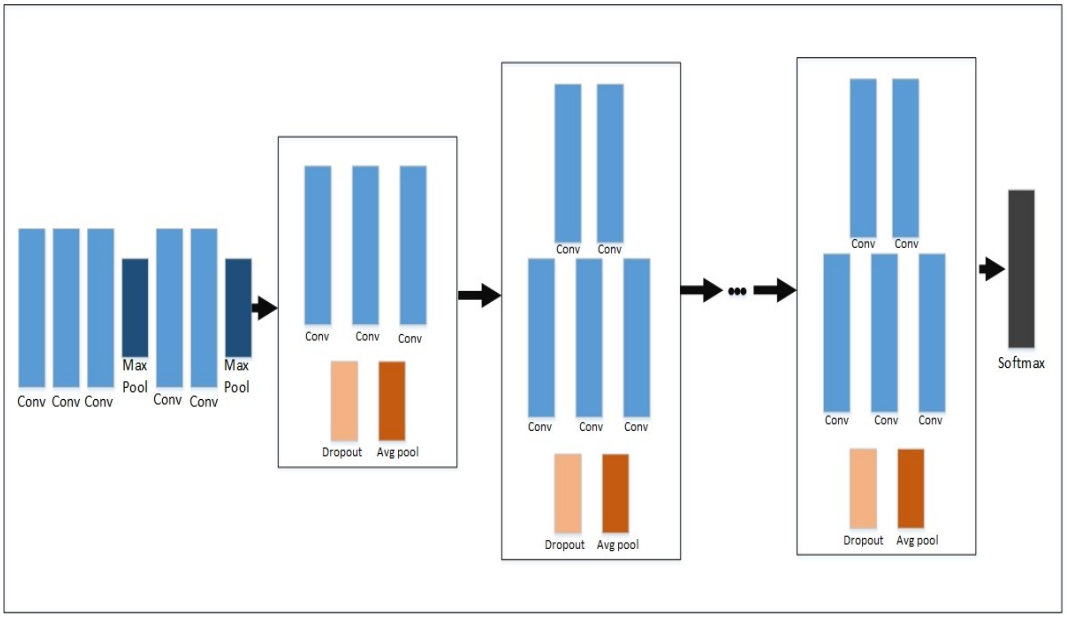


FIGURE 4.35: Inception V3 Architecture

4.5.4 LSTM-CNN Architecture

Standard Vanilla LSTM cannot be modeled for input with spatial structure such as images and videos.

The CNN Long Short-Term Memory Architecture was created to solve sequence prediction problems using spatial inputs like photos and videos. Convolutional Neural Network (CNN) layers for feature extraction on input data are paired with LSTMs to facilitate sequence prediction in the LSTM-CNN architecture.
LSTMs-CNN were developed for visual time series prediction problems and the application of generating textual descriptions from sequences of images (e.g. videos). Particularly, the problems of:

- **Activity Recognition:** It helps to generate a description of an activity as shown in a sequence of images.
- **Image Description:** It helps to generate a description of a single graphic image.
- **Video Description:** Creating a textual description of an image sequence.

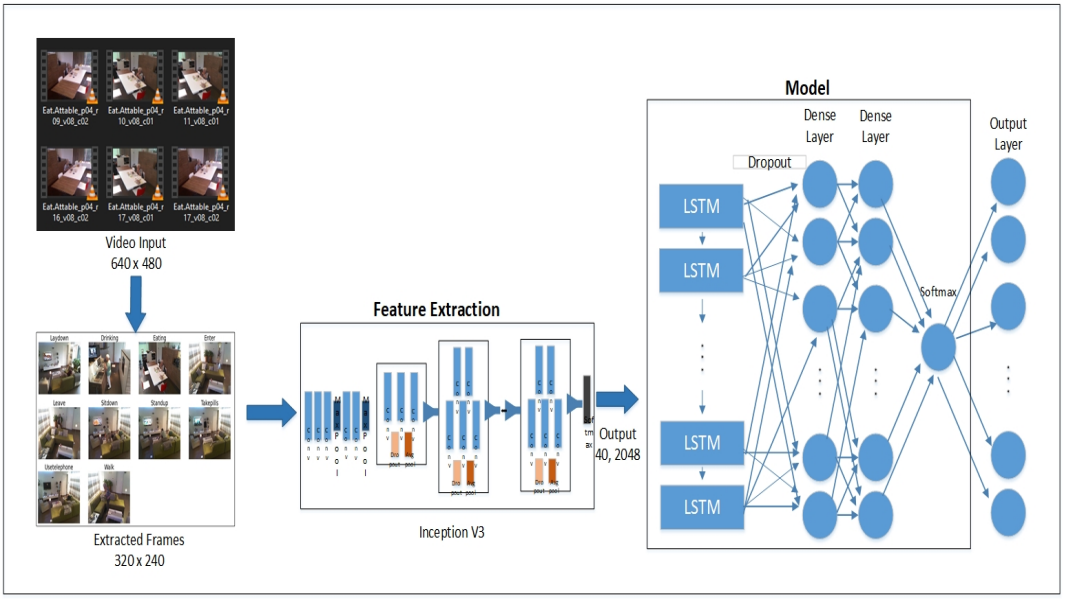


FIGURE 4.36: LSTM-CNN Architecture Diagram

The model employs lstm layers, which are followed by thick layers that are repeated three times. In the last layer, the Softmax activation function is applied. The model architecture is shown in Figure 4.36.

4.5.5 LSTM Layers

Due to its particular memory cells, LSTM is a type of RNN which has better outcomes in feature extraction of sequence or time-series data than simple CNN[96]. The temporal features in the sequence data are extracted by the two layers of LSTMs. Each layer of LSTM has 32 memory cells[97]. The inputs are sent to different gates, including input gates, forgetting gates and output gates, in order to control the behavior of each memory cell.

The activation of each LSTM unit is calculated by the following formula:

$$h_t = f(w_{i,h}.x_t + w_{h,h}.h_{t-1} + b) \tag{4.8}$$

where h_t and h_{t-1} are used to represent the activation at time t and t1, respectively, f is a non-linear activation function,$w_{i,h}$ is the input-hidden weight matrix, and $w_{h,h}$ is the hidden-hidden weight matrix, and lastly b is the hidden bias vector.
The output of the LSTM layer has three dimensions (samples, time steps, and the input dimension).

4.5.6 Convolutional and Pooling Layer

The convolutional layer is the most important unit in CNN, which uses convolution kernels to convolve the inputs. It works as a filter and is then activated by a non-linear function[98].
In the convolutional layers, the feature maps are calculated using rectified linear units (ReLU), and its non-linear function is defined as:

$$y = max(0, x) \tag{4.9}$$

Between the two convolutional layers is a max-pooling layer that handles the down-sampling operation. It has two main functions. One of which is to minimise the parameters while keeping dominating features, while the second is to filter the interference noise created by the unconscious jitter that humans perform while doing an action.

4.5.7 Training Model

The features are obtained from Inception-V3. These features are passed to the LSTM model. After that, some new layers are defined[99]. The new layers include dense layers with 'relu' activation function along with some dropouts between multiple layers. After combination of different layers are experimented with, a 2048-wide LSTM layer was determined followed by a 512 dense layer and a 0.5 dropout. Note that the features from a video are used as the input to the LSTM layers. Instead of passing features from each frame, the number of frames to be passed are reduced instead. This reduces over-fitting and speeds up the programme because fewer features are being processed. Features from approx. 30-50 frames per video are sent as sequence to the LSTM. At the final layer, 'softmax' activation is used, which is basically used to provide a class as output in terms of probability, i.e. it turns the input numbers to probabilities that sum to 1[36]. Thus, it outputs a probability distribution. "Categorical cross entropy" is used as a loss measure

and Adam optimizer is used with learning rate of 0.0001. Also, the metrics for training and testing are'accuracy' and 'top 5 categorical accuracy'. The model is compiled using these parameters and is trained[93]. Thus training is based on above mentioned architecture. The training and testing samples are divided according to the file provided in official website. This ensures uniform division of videos. The model is trained to learn in such a way to minimize the validation loss. When learning is stopped, i.e. the validation loss keeps on increasing, then the training is stopped. The accuracy metrics are used for evaluation. When doing the training, the model weights are saved whenever validation loss is decreased. When training is finished, the best weights of the model are saved and those weights are further used in classifying our own videos.

4.5.8 Results

The results show that training accuracy of 90% is achieved for the model used. To compare the result, different evaluation matrices are used including Accuracy, f1-score, recall, precision, auc and roc. The result is evaluated using the following matrices: Accuracy is the measure of correct classification of videos by the proposed system and it is defined as follows:

$$Accuracy = \frac{\text{TP} + \text{TN}}{\text{Total Sample in the dataset}}$$

Precision is the proportion of positive activity that have actually the said activity. It is calculated using the following:

$$Precision = \frac{\text{Correctly classified positive images}}{\text{Total classified positive images}}$$

F1-score is the harmonic mean of precision and recall and it is defined as

$$F_1 = 2\frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

Recall is the ability of proposed model to detect all the potential activities. It is calculated as follows:

$$Recall = \frac{\text{Correctly classified positive images}}{\text{Total positive images in the dataset}}$$

The results based on these metrics are shown in Table 4.6.

TABLE 4.6: Results of LSTM-CNN

	Precision	**Recall**	**F1-score**
Drinking	0.27	0.13	0.17
Eating	0.65	0.80	0.72
Enter	0.69	0.88	0.72
Leave	0.76	0.38	0.50
Lay	0.55	0.75	0.63
Sit	0.51	0.53	0.52
Stand	0.50	0.29	0.37
Take pills	0.58	0.41	0.48
Use Phone	0.43	0.59	0.50
Walk	0.41	0.62	0.50

Confusion matrix for the result is shown in Figure 4.36 with mapping used: ['Drink':0, 'Eat':1, 'Enter':2, 'Leave':3, 'Lay':4, 'Sit':5, 'Stand':6, 'pills':7, 'phone':8, 'Walk':9]

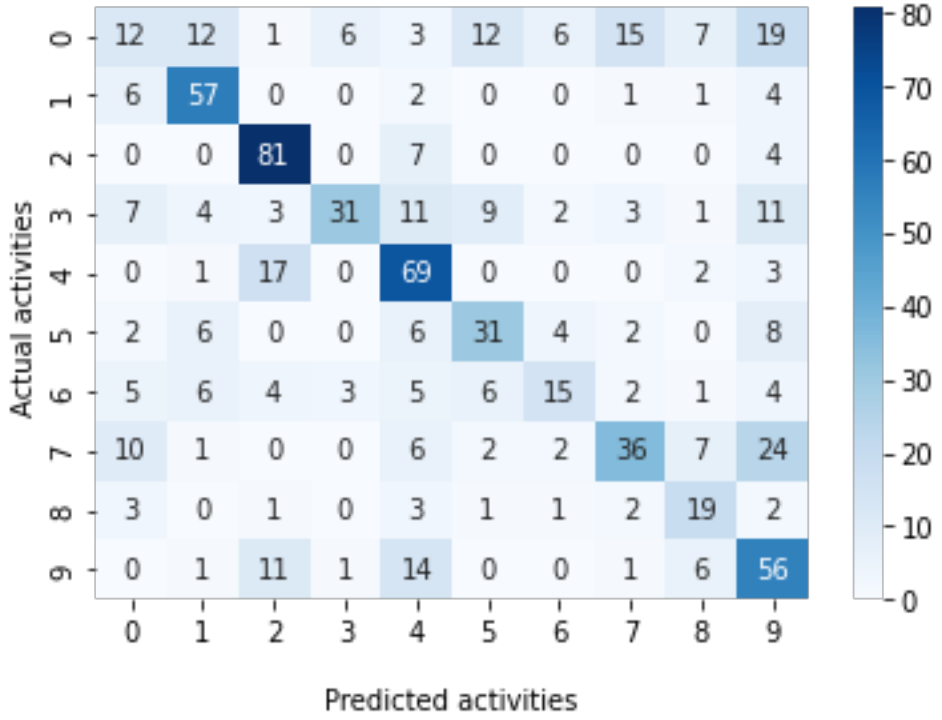


FIGURE 4.37: Confusion Matrix

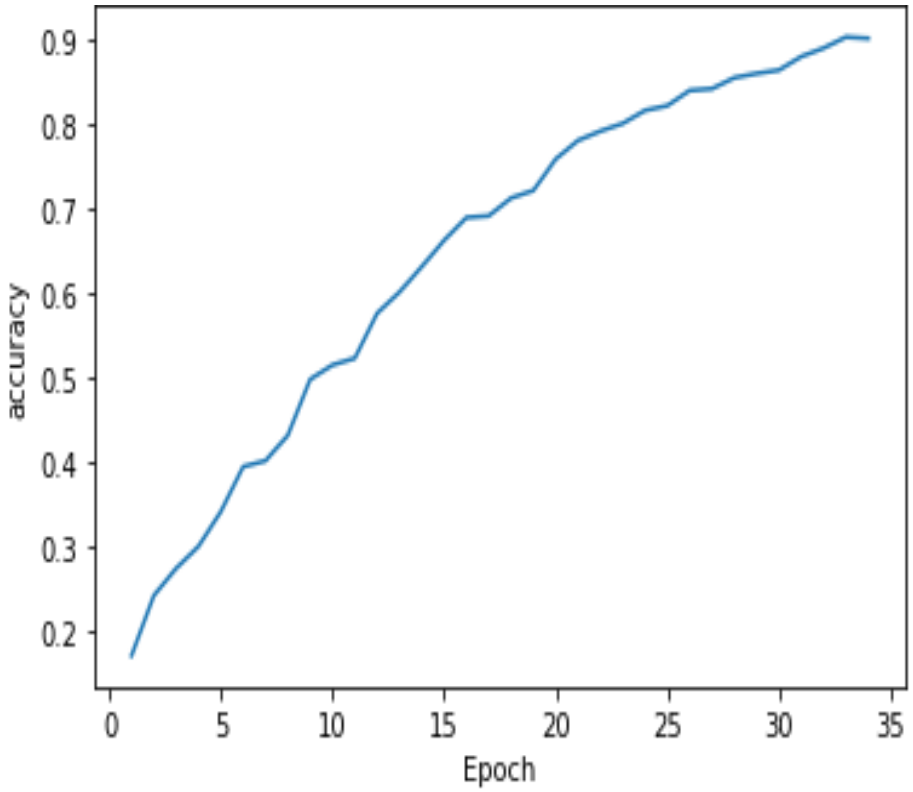


FIGURE 4.38: Training Accuracy

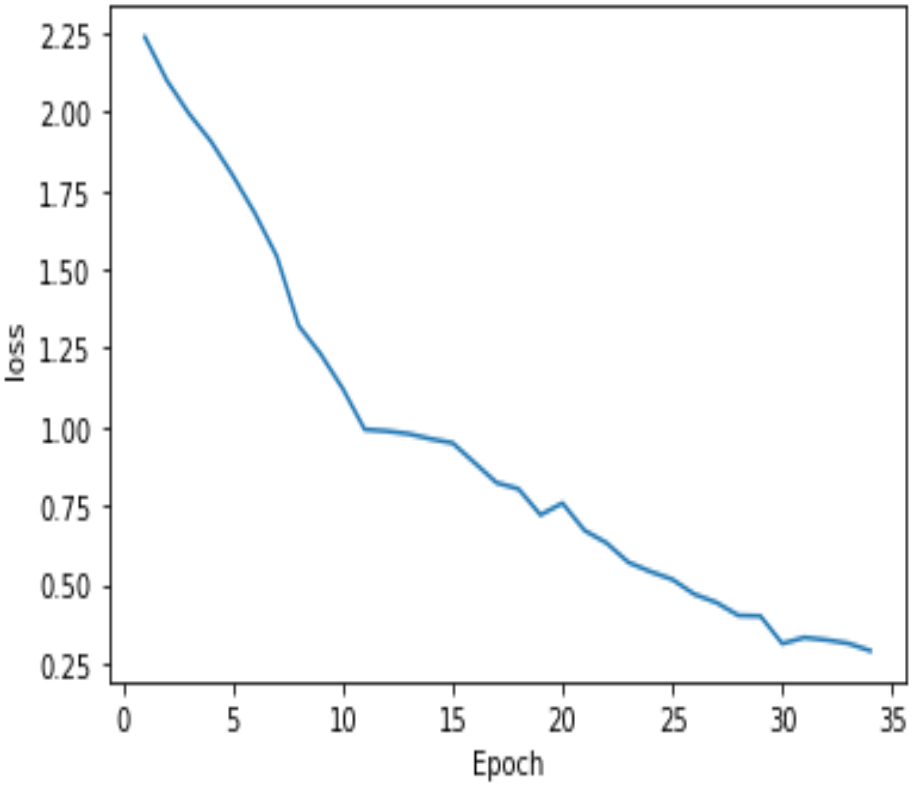


FIGURE 4.39: Training Loss

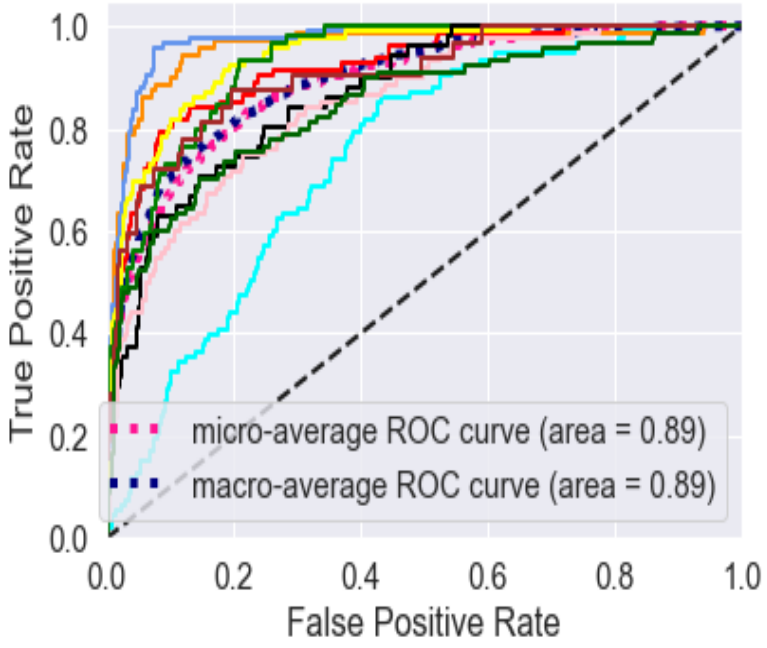


FIGURE 4.40: ROC Curve

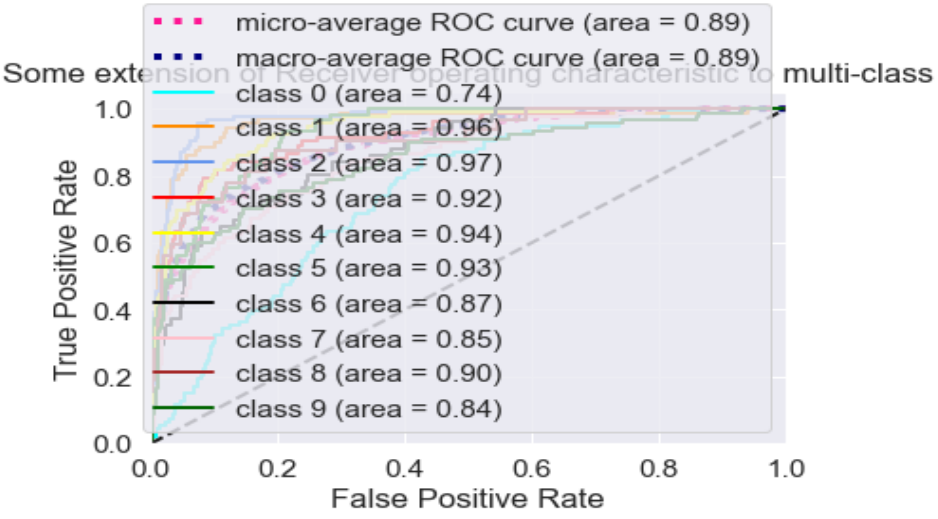


FIGURE 4.41: Labels for ROC

4.6 Predictive Modeling

The results from activity recognition are stored in the database and used in the next step of predicting activities. Prediction in a dynamic situation requires modeling relationships among events instead of just static objects[100]. Although correlations might be useful in some situations, causality is necessary to accurately characterise spatio-temporal interactions. Using previous data, predictive analytics algorithms are utilised to estimate future events. Historical data is utilised to create a mathematical model that captures relevant trends for future investigation. This predictive model is then used to current data in order to forecast what will

happen next or to advise actions to take in order to achieve the best results. Objectivity is critical, and it is achieved through data collection and analysis. Data collection is tailored to the aged in question's daily routine. LSTM is used here for predictions of the next events about to take place. LSTMs have been previously explained in section 4.4.3. The state of a LSTM network is represented through a state space vector. This technique allows to keep tracks of dependencies of new observations with those of the past ones(even the very far ones).

4.6.1 Data preparation

The series of observations is broken down into several examples for the LSTM to learn from. For the one-step prediction that is being learned, the sequence is broken into numerous input/output patterns called samples, with three time steps as input and one time step as output.

4.6.2 Stacked LSTM

Stacked LSTM model is made up of many hidden LSTM layers stacked one on top of the other. An LSTM layer requires a three-dimensional input and LSTMs produce an outpur of two-dimensional output as an explanation from the end of the sequence by default The return_sequences=True argument is set as true on the layer to solve the problem and the LSTM is made to have output a value for each time step in the input data. This allows the model to have 3D output from hidden LSTM layer as input to the next layer.

4.6.3 Results

The results for the model are given in Figure 4.42. To compare the result, different evaluation matrices are used. The result is evaluated using the following matrices: Accuracy is the measure of correct classification of videos by the proposed system and it is defined as follows:

$$Accuracy = \frac{\text{TP} + \text{TN}}{\text{Total Sample in the dataset}}$$

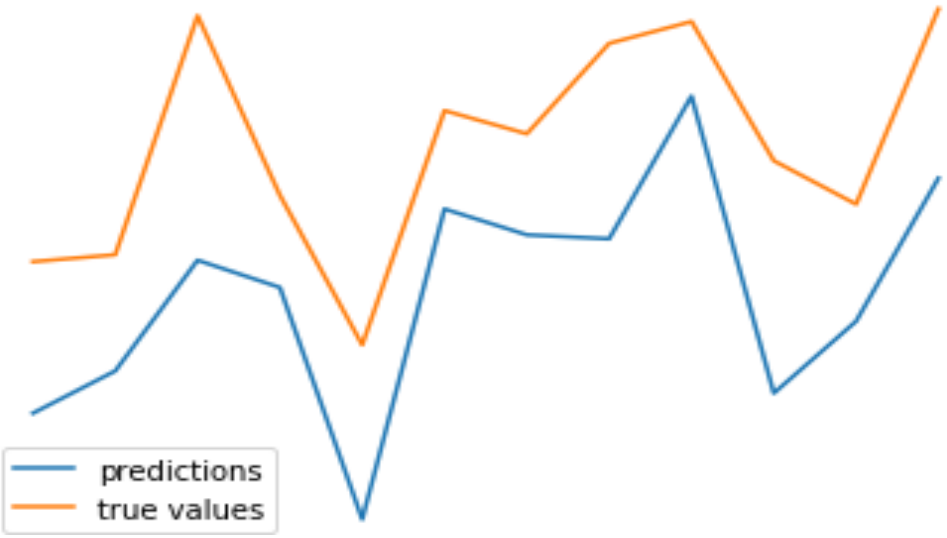


FIGURE 4.42: Result for predictive model

4.7 Anomaly detection using an Autoencoder

The anomaly is defined as any kind of behaviour which deviates from defined norms. In this work, anomaly is defined as any behaviour or event which deviates significantly from the predicted values in the previous section.

4.7.1 Preparing data

Data from the previous section is ordered and timestamped. Data values are taken from the training timeseries data file and then normalized. There are values for every 5 mins for 14 days.

4.7.2 Model

Sequences of the time series to be used for training, are created by combining the contiguous data values from the training data. A convolutional reconstruction autoencoder model is built for training the dataset. The input is sent to the model of shape (batch_size, sequence_length, num_features) and returns output of the same shape. The anomalies are detected by determining how well the model can reconstruct the input data. In order to achieve this, MAE loss on training samples is found. Then the max MAE loss value is found. This value determines the worst that the model has performed while trying to reconstruct a sample. This value is then selected as the threshold for anomaly detection. If the reconstruction loss for a sample is greater than this threshold value then it is inferred that the model is seeing a pattern that it isn't familiar with. This sample is labelled as an anomaly.

4.7.3 Results

The model is trained and the results are shown below.

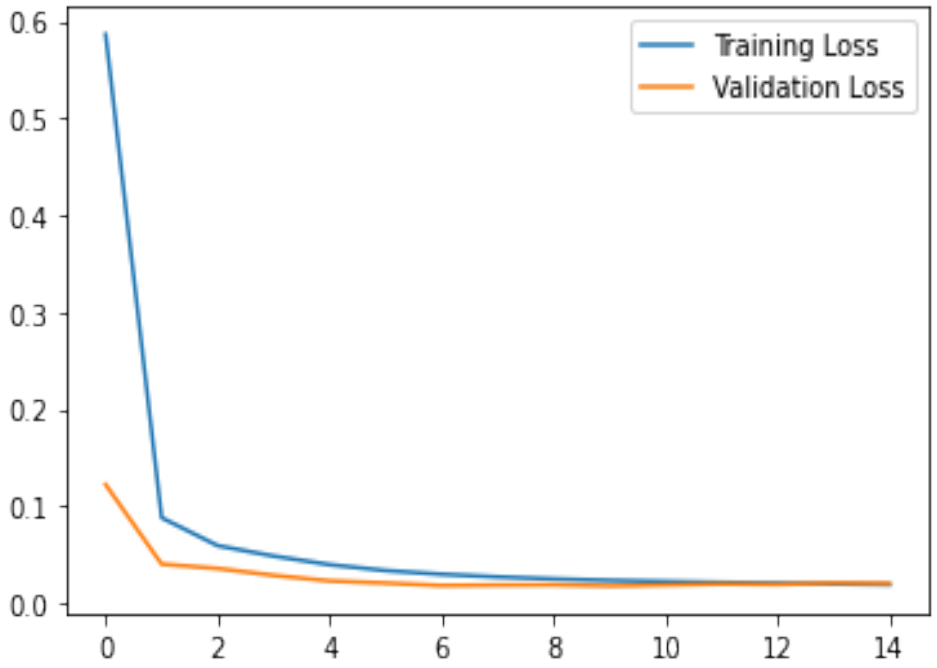


FIGURE 4.43: Training accuracy and loss

The samples of the data which are anomalies are known. With this, the results for anomalies are shown in Figure 4.44.

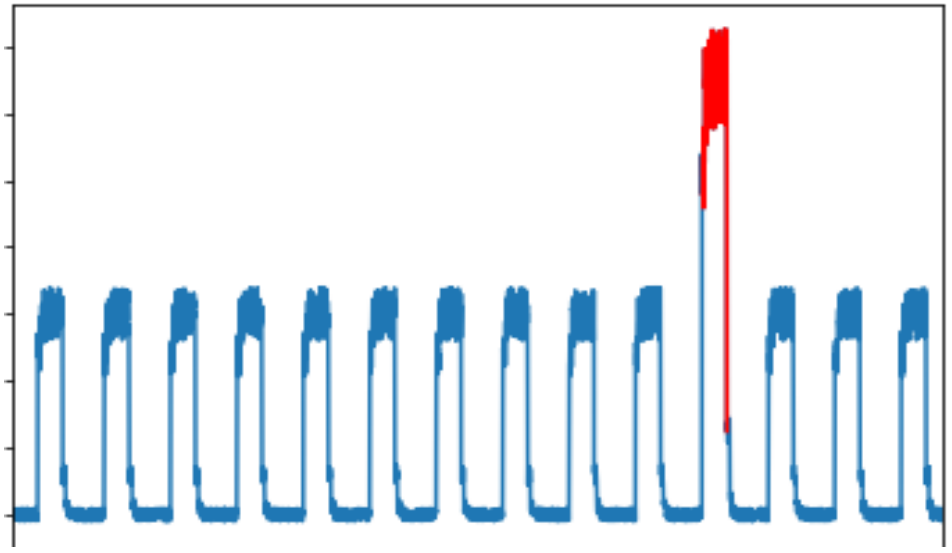


FIGURE 4.44: Anomaly Detection

4.8 Trend analysis

The sleep pattern trend is analysed by collecting sleep data over a period of 2 weeks. The timestamps are taken from the dataset from sleep start and sleep end period. The database is crawled and the timestapms for the sleep period over two weeks are taken and stored in a file. Timestamps are stored as dictionary of pairs

and it is a chronologically ordered list. An example of sleeping hours for a single person over different days is shown in Figure 4.45.

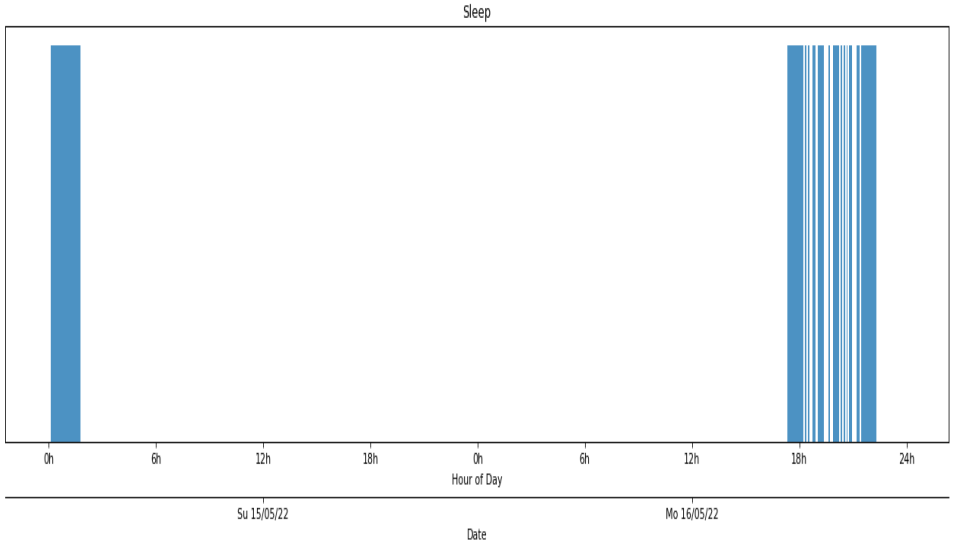


FIGURE 4.45: Sleeping hours over different days

In the next step, binary maps are created representing the sleep status of the person. This creates timestamps of every 5 minutes for 24 hours, creating a total of 289 bins for a day. The graph in Figure 4.46 shows the number of different people asleep at different times. Gaussian smoothing is applied on each curve for better visualization results.

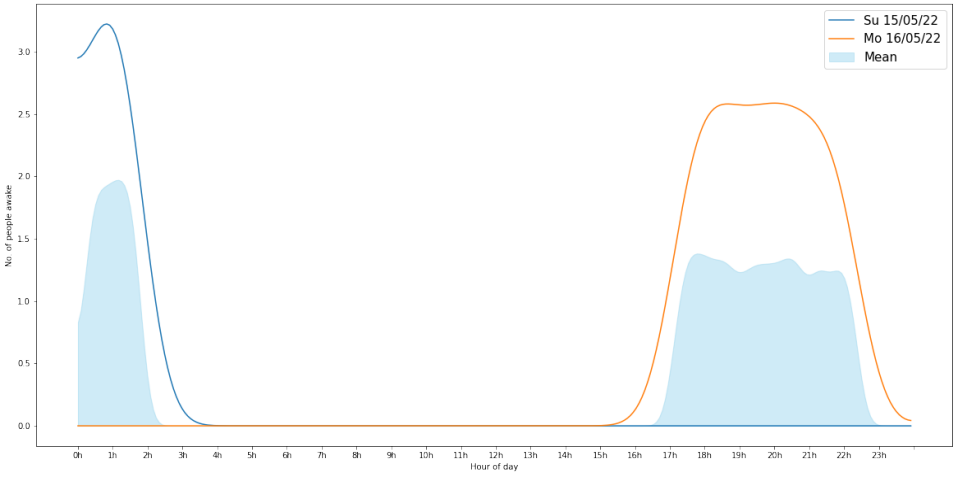


FIGURE 4.46: No. of people awake at different times

The Auto Regression Model is used for time series analysis. In order to do this, sequencce to sequence model is used with one encoder layer and one decoder layer. The model is fit on the dataset, the forecast is predicted and the seasonalities are drawn. After the model is fit, predict method is called to obtain the forecast.

4.8.1 Results

The trend component is shown in Figure 4.47

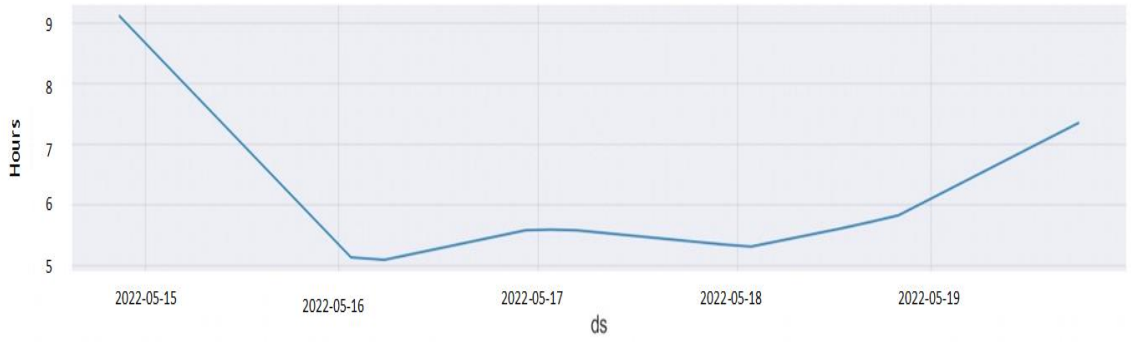


FIGURE 4.47: Trend for sleep

Chapter 5

Conclusion

5.1 Conclusion

The project focuses on activity detection of elderly activities and comparing the results using different models. In addition, the results using simple RNN, LSTM and hybrid LSTM+CNN are compared.

When compared to other models, the findings obtained in the preceding section reveal that the hybrid model of Convolutional neural network with LSTM has a high accuracy for activity detection. The findings suggest that emergency situations may be identified with great accuracy, allowing older people to live independently. They can be monitored remotely and the alarm can be generated.

We further contribute to the literature by providing a novel hybrid deep learning model architecture with a reliable output for detecting activities. This model could be used to detect further detect other related activities by training it on the relevant dataset. Hence, there is less need to change the model as values can be revaluated as soon as additional information and dataset presents itself. This research also contributes to the deep learning literature by explaining the differences that results from using different model architectures. The findings might assist to lower the total cost of relocating an older person to an old age home by allowing them to live freely in their own houses.

5.1.1 Future Work

Further research might examine a more wide range of activities including heart attack and insomnia based on the need and the dataset available. The data can also be collected in relevance to the patient and then the model can be trained in order to tune the model according to the need of the patient. In addition, it might be interesting to focus on what is most important, either daily activities or the emergency situation and collect the data accordingly. As an example, the activities for a patient of dementia can be different from those of a patient of

Parkinson's disease. The model can also be trained according to the suggestions of the patient's doctor.

In addition, some theoretical direction for further research might focus on other models that can be used for activity detection such as Convolutional Autoencoders and Transformers. The results can then be compared to check the performance of these methodologies. Different models provide different results therefore it might also be interesting to combine different methods and check results. By examining the activities in a given period, the results can also be analyzed for pattern recognition of sleep behavior and eating behaviour of the patient. Similarly, anomalies in the patterns can be detected and the alarm can be generated based on the type of anomaly. The dataset can also be extended to include a wide range of camera types, and lighting settings in order to detect the activities accurately in different weather and light conditions. All of these suggestions can be worked on to improve daily activity detection of elderly.

References

[1] Sumit Majumdera. Smart Homes for Elderly Healthcare—Recent Advances and Research Challenges. *Sensors (Basel)*, 2017.

[2] Maria Ahmed Qureshi. Independent Living for Persons with Disabilities and Elderly People Using Smart Home Technology. *Neural Comput Applic*, 2021.

[3] Roschelle L. Fritz. A Nurse-Driven Method for Developing Artificial Intelligence in "Smart" Homes for Aging-in-Place. *Nurs Outlook*, 2018.

[4] Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, and Michael Littman. Activity recognition from accelerometer data. *AAAI*, 3:1541–1546, 01 2005.

[5] Daniel Flores-Martin. Smart Nursing Homes: Self-Management Architecture Based on IoT and Machine Learning for Rural Areas. *Wireless Communications and Mobile Computing for Ambient Assisted Living*, 2021:15, 2021.

[6] World Bank. Disability Inclusion. 2021.

[7] Kiwon Yun, Jean Honorio, Debaleena Chattopadhyay, Tamara Berg, and Dimitris Samaras. Two-person interaction detection using body-pose features and multiple instance learning. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 28–35, 06 2012.

[8] Abdur Rahim Mohammad Forkan, Philip Branch, Prem Prakash Jayaraman, and Andre Ferretto. An internet-of-things solution to assist independent living and social connectedness in elderly. *Trans. Soc. Comput.*, 2(4), dec 2019.

[9] Matthew Lee and Anind Dey. Reflecting on pills and phone use: Supporting awareness of functional abilities for older adults. *Conference on Human Factors in Computing Systems - Proceedings*, pages 2095–2104, 05 2011.

[10] Miguel Ángel Antón. Non-Invasive Ambient Intelligence in Real Life: Dealing with Noisy Patterns to Help Older People. *MDPI*, page 19, 2019.

[11] Isabelle Guyon and André Elisseeff. An introduction of variable and feature selection. *J. Machine Learning Research Special Issue on Variable and Feature Selection*, 3:1157 – 1182, 01 2003.

[12] U. Maurer, A. Smailagic, D.P. Siewiorek, and Michael Deisher. Activity recognition and monitoring using multiple sensors on different body positions. *Proc Int Workshop Wearable Implantable Body Sensor Netw*, 2006:4 pp.–, 05 2006.

[13] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Mining actionlet ensemble for action recognition with depth cameras. pages 1290–1297, 2012.

[14] June-Goo Lee, Sanghoon Jun, Younghoon Cho, Hyunna Lee, Guk Bae Kim, Joon Beom Seo, and Namkug Kim. Deep learning in medical imaging: General overview. *Korean Journal of Radiology*, 18:570 – 584, 2017.

[15] Dana Rezazadegan, Sareh Shirazi, Michael Milford, and Ben Upcroft. Evaluation of object detection proposal under condition variations. 06 2015.

[16] Yun Cheong and Wei Chew. The application of image processing to solve occlusion issue in object tracking. *MATEC Web of Conferences*, 152:03001, 01 2018.

[17] Kusuyama T. Fukuda K. Uchikawa K. Morimoto, T. Human color constancy based on the geometry of color distributions. *Journal of vision*, 21:7, 2021.

[18] Mary Bravo and Hany Farid. Object recognition in dense clutter. *Perception psychophysics*, 68:911–8, 09 2006.

[19] Shuyuan Xu, Jun Wang, Wenchi Shou, Tuan Ngo, Abdul-Manan Sadick, and Xiangyu Wang. Computer vision techniques in construction: A critical review. *Archives of Computational Methods in Engineering*, 28, 10 2020.

[20] vic logo. Data collection challenges and improvements. https://www.vic.gov.au/victorian-family-violence-data-collection-framework/data-collection-challenges-and-improvements/. Accessed: 2022-01-30.

[21] Ethan Ace. Resolving IP Camera / VMS Time Sync Problems. https://ipvm.com/reports/network-time-sync-issues. Accessed: 2022-01-30.

[22] Bassel Chawky, Ahmed Samir Roshdy, A. Ali, and Howida Shedeed. *A Study of Action Recognition Problems: Dataset and Architectures Perspectives*, pages 409–442. 01 2018.

[23] Jeffrey Lockhart and Gary Weiss. Limitations with activity recognition methodology data sets. 09 2014.

[24] Julian Shin Choonsung Dey Anind K. Hong, Jin Hyuk Ramos. An activity recognition system for ambient assisted living environments. *Communications in Computer and Information Science*, 362:11, 2013.

[25] Liang Wang Yong Du, Wei Wang. Hierarchical Recurrent Neural Network for Skeleton Based Action Recognition. *CVPR*, page 9, 2015.

[26] Charith Pal, Sandipan Abhayaratne. Video-based activity level recognition for assisted living using motion features. *ACM International Conference Proceeding Series*, page 6, 2015.

[27] Daniele Liciotti, Michele Bernardini, L. Romeo, and E. Frontoni. A sequential deep learning application for recognising human activities in smart homes. *Neurocomputing*, 396:501–513, 2020.

[28] Martin Kampel Rainer Planinc. Computer Vision for Active and Assisted Living. 2016.

[29] Fernando Moya Rueda and Gernot A. Fink. Learning attribute representation for human activity recognition. *CoRR*, abs/1802.00761, 2018.

[30] Haoyu Li, Stéphane Derrode, and Wojciech Pieczynski. An adaptive and on-line IMU-based locomotion activity classification method using a triplet Markov model. *Neurocomputing*, (362):94 – 105, October 2019.

[31] Ronald Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976–990, 2010.

[32] Gianluigi Ciocca Marco Buzzelli, Alessio Albé. A Vision-Based System for Monitoring Elderly People at Home. 2020.

[33] Majid Ali Quaid and Ahmad Jalal. Wearable sensors based human behavioral pattern recognition using statistical features and reweighted genetic algorithm. *Multimedia Tools and Applications*, 79:1–23, 03 2020.

[34] Vito Marani Roberto Nitti Massimiliano D'Orazio Tiziana Stella Ettore Mosca, Nicola Renò. Human walking behavior detection with a RGB-D sensors network for ambient assisted living applications. *CEUR Workshop Proceedings*, 2061:13, 2018.

[35] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. 25, 2012.

[36] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. pages 770–778, 2016.

[37] Damiano Malafronte Philipe A. Dias. Gaze Estimation for Assisted Living . 2020.

[38] Pau Climent-Pérez. A review on video-based active and assisted living technologies for automated lifelogging . 2019.

[39] Eric McAdams, Claudine Gehin, Norbert Noury, Carolina Ramon, Ronald Nocua, Bertrand Massot, Aurélien Oliveira, André Dittmar, Chris Nugent, and Jim McLaughlin. *Biomedical Sensors for Ambient Assisted Living*, volume 55, pages 240–262. 01 1970.

[40] Zelun Luo. Computer Vision-Based Descriptive Analytics of Seniors' Daily Activities for Long-Term Health Monitoring. *Proceedings of Machine Learning Research*, 2018.

[41] Stefano Beltrami Giorgio Schmid Micaela Guerra, Bruna Maria Vittoria Ramat. Automatic Pose Recognition for Monitoring Dangerous Situations in Ambient-Assisted Living. *Frontiers in Bioengineering and BiotechnologyS*, 8:12, 2020.

[42] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017.

[43] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015.

[44] João Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *CoRR*, abs/1808.01340, 2018.

[45] Lucas Smaira, João Carreira, Eric Noland, Ellen Clancy, Amy Wu, and Andrew Zisserman. A short note on the kinetics-700-2020 human action dataset. *CoRR*, abs/2010.10864, 2020.

[46] Ang Li, Meghana Thotakuri, David A. Ross, João Carreira, Alexander Vostrikov, and Andrew Zisserman. The ava-kinetics localized human actions video dataset. *CoRR*, abs/2005.00214, 2020.

[47] Hang Zhao, Antonio Torralba, Lorenzo Torresani, and Zhicheng Yan. Hacs: Human action clips and segments dataset for recognition and temporal localization. pages 8667–8677, 10 2019.

[48] Quan Kong, Ziming Wu, Ziwei Deng, Martin Klinkigt, Bin Tong, and Tomokazu Murakami. Mmact: A large-scale dataset for cross modal human action understanding. pages 8657–8666, 10 2019.

[49] Nishant Rai, Haofeng Chen, Jingwei Ji, Rishi Desai, Kazuki Kozuka, Shun Ishizaka, Ehsan Adeli, and Juan Carlos Niebles. Home action genome: Cooperative compositional action understanding. *CoRR*, abs/2105.05226, 2021.

[50] Silvio Giancola, Mohieddine Amine, Tarek Dghaily, and Bernard Ghanem. Soccernet: A scalable dataset for action spotting in soccer videos. *CoRR*, abs/1804.04527, 2018.

[51] Arka Sadhu, Tanmay Gupta, Mark Yatskar, Ram Nevatia, and Aniruddha Kembhavi. Visual semantic role labeling for video understanding. *CoRR*, abs/2104.00990, 2021.

[52] Yujia Qin, Jiajie Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Elle: Efficient lifelong pre-training for emerging data, 2022.

[53] Yizhak Ben-Shabat, Xin Yu, Fatemeh Sadat Saleh, Dylan Campbell, Cristian Rodriguez Opazo, Hongdong Li, and Stephen Gould. The IKEA ASM dataset: Understanding people assembling furniture through actions, objects and pose. *CoRR*, abs/2007.00394, 2020.

[54] WISDM Lab. Wisdm: Wireless sensor data mining. https://www.cis.fordham.edu/wisdm/dataset.php, Dec. 02, 2012 [Online]. Accessed: 2022-05-07.

[55] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkl, Florian Wagner, Alois Ferscha, Jakob Doppler, Clemens Holzmann, Marc Kurz, Gerald Holl, Ricardo Chavarriaga, Marco Creatura, Jose del R. Millan, and R An. Walk-through the opportunity dataset for activity recognition in sensor rich environments. 01 2010.

[56] Amir R. Zamir b Yu-Gang Jiangc Alex Gorbane Ivan Laptev d Rahul Sukthankar e Mubarak Shahag Haroon Idrees a, . The THUMOS challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, page 23, 2016.

[57] Mi Zhang and Alexander Sawchuk. Usc-had: a daily activity dataset for ubiquitous activity recognition using wearable sensors. pages 1036–1043, 09 2012.

[58] Piero Zappi, Clemens Lombriser, Thomas Stiefmeier, Elisabetta Farella, Daniel Roggen, Luca Benini, and Gerhard Tröster. *Activity Recognition from On-Body Sensors: Accuracy-Power Trade-Off by Dynamic Sensor Selection*, pages 17–33. 01 1970.

[59] Attila Reiss and Didier Stricker. Introducing a new benchmarked dataset for activity monitoring. 06 2012.

[60] Marc Bächlin, Meir Plotnik, Daniel Roggen, Nir Giladi, Jeffrey Hausdorff, and G Tröster. A wearable system to assist walking of parkinson´s disease patients. *Methods of information in medicine*, 49:88–95, 12 2009.

[61] Rose LAB. Rose NTU Dataset. https://rose1.ntu.edu.sg/dataset/actionRecognition/, Jan. 12, 2016 [Online]. Accessed: 2022-05-21.

[62] Rui Dai, Srijan Das, Saurav Sharma, Luca Minciullo, Lorenzo Garattoni, Francois Bremond, and Gianpiero Francesca. Toyota smarthome untrimmed: Real-world untrimmed videos for activity detection, 2020.

[63] Toyota. Toyota dataset. https://project.inria.fr/toyotasmarthome/, Oct. 01, 2019 [Online]. Accessed: 2022-01-30.

[64] Michal Kepski. Wisdm: Wireless sensor data mining. https://www.cis.fordham.edu/wisdm/dataset.php, Apr. 18, 2017 [Online]. Accessed: 2022-05-07.

[65] Oresti Banos, Rafael García, Juan Holgado-Terriza, Miguel Damas, Hector Pomares, Ignacio Rojas, Alejandro Saez, and Claudia Villalonga. mhealthdroid: A novel framework for agile development of mobile health applications. volume 8868, pages 91–98, 12 2014.

[66] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Prentow, Mikkel Kjærgaard, Anind Dey, Tobias Sonne, and Mads Jensen. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. pages 127–140, 11 2015.

[67] Kerem Altun, Billur Barshan, and Orkun Tunçel. Comparative study on classifying human activities with miniature inertial and magnetic sensors. *Pattern Recognition*, 43:3605–3620, 10 2010.

[68] Oresti Banos, Mate Attila Toth, Miguel Damas, Hector Pomares, and Ignacio Rojas. Dealing with the effects of sensor displacement in wearable activity recognition. *Sensors*, 14(6):9995–10023, 2014.

[69] Ugur Demir, Yogesh S. Rawat, and Mubarak Shah. Tinyvirat: Low-resolution video action recognition. *CoRR*, abs/2007.07355, 2020.

[70] Zhe Ren, Junchi Yan, Bingbing Ni, Bin Liu, Xiaokang Yang, and Hongyuan Zha. Unsupervised deep learning for optical flow estimation. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[71] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017.

[72] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015.

[73] Étienne Mémin and Patrick Pérez. Hierarchical estimation and segmentation of dense motion fields. *International Journal of Computer Vision*, 46:129–155, 02 2002.

[74] Shang-Hong Lai and Baba Vemuri. Reliable and efficient computation of optical flow. *International Journal of Computer Vision*, 29:87–105, 08 1998.

[75] Nurullah Calik, Mehmet Ali Belen, and Peyman Mahouti. Deep learning base modified mlp model for precise scattering parameter prediction of capacitive feed antenna. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, 33(2):e2682, 2020.

[76] Yuanyuan Wei, Julian Jang-Jaccard, Fariza Sabrina, Amardeep Singh, Wen Xu, and Seyit Camtepe. Ae-mlp: A hybrid deep learning approach for ddos detection and classification. *IEEE Access*, 9:146810–146821, 2021.

[77] M.J. Black and A.D. Jepson. Estimating optical flow in segmented images using variable-order parametric models with local deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):972–986, 1996.

[78] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE Multimedia - IEEEMM*, 19:4–10, 02 2012.

[79] John Barron, David Fleet, and S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12:43–77, 02 1994.

[80] Annalisa Franco, Antonio Magnani, and Dario Maio. A multimodal approach for human activity recognition based on skeleton and rgb data. *Pattern Recognit. Lett.*, 131:293–299, 2020.

[81] Ahmad Jalal, Yeon-Ho Kim, Yong-Joong Kim, Shaharyar Kamal, and Daijin Kim. Robust human activity recognition from depth video using spatiotemporal multi-fused features. *Pattern Recogn.*, 61(C):295–308, jan 2017.

[82] Adnan Farooq, Ahmad Jalal, and Shaharyar Kamal. Dense rgb-d map-based human tracking and activity recognition using skin joints features and self-organizing map. 07 2018.

[83] Francesco Piccialli, Vittorio Di Somma, Fabio Giampaolo, Salvatore Cuomo, and Giancarlo Fortino. A survey on deep learning in medicine: Why, how and when? *Inf. Fusion*, 66:111–137, 2021.

[84] Robert Hecht-Nielsen. Theory of the backpropagation neural network. *International 1989 Joint Conference on Neural Networks*, pages 593–605 vol.1, 1989.

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

[86] Yang Zhao and Neal Patwari. Noise reduction for variance-based device-free localization and tracking. pages 179–187, 06 2011.

[87] Sridevi Durga, Rishabh Nag, and Esther Daniel. Survey on machine learning and deep learning algorithms used in internet of things (iot) healthcare. In *2019 3rd international conference on computing methodologies and communication (ICCMC)*, pages 1018–1022. IEEE, 2019.

[88] Pramila P Shinde and Seema Shah. A review of machine learning and deep learning applications. In *2018 Fourth international conference on computing communication control and automation (ICCUBEA)*, pages 1–6. IEEE, 2018.

[89] AK Sampath and Dr N Gomathi. Decision tree and deep learning based probabilistic model for character recognition. *Journal of Central South University*, 24(12):2862–2876, 2017.

[90] Nabil Zerrouki and Amrane Houacine. Combined curvelets and hidden markov models for human fall detection. *Multimedia Tools and Applications*, 77:6405–6424, 2017.

[91] Alex Krizhevsky Ilya Sutskever Ruslan Salakhutdinov Nitish Srivastava, Geoffrey Hinton. Dropout: A simple way to prevent neural networks from overfitting.

[92] Michael Avendi. Video classification with cnn, rnn, and pytorch | by michael avendi | how to ai | medium. [Online; accessed 2022-05-06].

[93] Junliang Xing Wenjun Zeng Wentao Zhu, Cuiling Lan. Co-occurrence Feature Learning for Skeleton based Action Recognition using Regularized Deep LSTM Networks. *The 30th AAAI Conference on Artificial Intelligence*, page 30, 2016.

[94] Palash Sharma. Keras LSTM Layer Explained for Beginners with Example. https://machinelearningknowledge.ai/keras-lstm-layer-explained-for-beginners-with-example/, Feb. 01, 2021 [Online]. Accessed: 2022-01-30.

[95] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. June 2014.

[96] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. December 2015.

[97] Hong-Bo Zhang, Yi-Xiang Zhang, Bineng Zhong, Qing Lei, Lijie Yang, Ji-Xiang Du, and Duan-Sheng Chen. A comprehensive survey of vision-based human action recognition methods. *Sensors*, 19(5), 2019.

[98] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. pages 2818–2826, 2016.

[99] Fouzi Harrou, Nabil Zerrouki, Dairi Abdelkader, Ying Sun, and Amrane Houacine. Automatic human fall detection using multiple tri-axial accelerometers. pages 74–78, 09 2021.

[100] Le Cun Yan, B Yoshua, and H Geoffrey. Deep learning. *nature*, 521(7553):436–444, 2015.