: N° d'ordre
: N° de série

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research

UNIVERSITY OF ECHAHID HAMMA LAKHDAR - EL OUED

FACULTY OF EXACT SCIENCES
Computer Science Department

Thesis
submitted in partial fulfilment of the requirements for the degree of

ACADEMIC MASTER

Field: **Mathematics and Computer Science**
Option: **Computer Science**
Specialty: **Distributed Systems and Artificial Intelligence**

Presented by :

- Fadhel Kamel
- Adaika Salem

Title

Fall detection for elderly-people monitoring using Deep Learning

Defended on June 16 th 2022

Examination Commitee :

M.	Nedioui Abd Elhamid	MAA	Chair
M.	Yagoub Mohamed Amine	MCB	Examinator
M.	Berdjouh Chafik	MAA	Supervisor

Academic year: 2021-2022

Acknowledgements

First of all, we thank "God" for giving us the patience, health and courage to do this work.

Today, we would like to extend our thanks and gratitude to our supervisor, "Mr. Berdjouh Chafik", who gave us and continues to give us many of his ideas and efforts without waiting for praise or thanks.

We can only thank the president of the Department of Computer Science, Professor "Mr. Medileh Saci". We also extend our sincere thanks to the members of the jury who honored us by accepting the verdict on our work. We also do not forget, of course, our professors at all levels of the university, to offer them our best wishes.

Finally, we would like to thank all those who, from near or far, have contributed with their encouragement, advice and support for us to do this work.

Fadhel Kamel.

Adaika Salem.

Dedications

First, we thank God Almighty for enabling us to complete this project. and giving us patience and determination to arrive on this day to reap the results of our efforts.

I dedicate the results of these efforts and success to my beloved parents and wish them continued health and wellness.

To all the honorable family of brothers and sisters.

To everyone who supported me and stood by my side.

To my partner in this project and my dear friend Salem.

To all the professors and students of the Department of Computer Science, especially students of the 2nd year master promotion 2022.

Fadhel Kamel.

Dedications

First of all, I thank ALLAH the Almighty for giving me the courage and patience to carry out this work despite all the difficulties encountered.
I thank my supervisor "Mr. Berdjouh Chafik" for his advice and skills, which helped us a lot despite his workload.
First of all I dedicate this modest work To my very dear mother who encouraged me in every wayWhile doing this work. And To my dear father, may God protect him. I also dedicate it to my brothers and sisters, The source of my joy and happiness.
To all my friends, especially my dear friend Kamel, who shared with me the trouble of preparing this work, and to all his family.
To all students of the Computer Science faculty, especially students of the 2nd year master promotion 2022.

Adaika Salem.

Abstract

Falls are one of the biggest threats to older adults, and they have significant psychological, physical, and financial implications. It is the leading cause of serious injuries, disabilities, hospitalizations and even death, especially if they do not receive urgent medical assistance. Response time greatly affects the effectiveness of medical intervention and reduces harm to the injured. As part of helping the elderly to lead their normal lives at home, smart monitoring systems have been proposed that detect falls and give alerts to health care providers. In this work, we present an instance of this system, which attempts to address the problem of fall detection through Vision-Based Deep Learning (DL) techniques. The proposed methodology uses Motion History Image (MHI) algorithm to determine the movement of a person through a set of frames in a (RGB) video clip. Using a Convolutional Neural Network (CNN) we will create an image classification model to detect the state of falling and not falling. The goal of the work is to correctly detect falls and reduce the number of false alarms for the system. Whereas, the accuracy rate of our model reached 95.61%.

Keywords : Fall Detection, Classification, Deep Learning, Vision-Based.

Résumé

Les chutes sont l'une des plus grandes menaces pour les personnes âgées et elles ont d'importantes répercussions psychologiques, physiques et financières. C'est la principale cause de blessures graves, d'invalidités, d'hospitalisations et même de décès, surtout s'ils ne reçoivent pas d'assistance médicale d'urgence. Le temps de réponse affecte grandement l'efficacité de l'intervention médicale et réduit les dommages causés aux blessés. Afin d'aider les personnes âgées à mener une vie normale à domicile, des systèmes de surveillance intelligents ont été proposés qui détectent les chutes et alertent les prestataires de soins de santé. Dans ce travail, nous présentons une instance de ce système, qui tente de résoudre le problème de la détection des chutes grâce à des techniques d'apprentissage en profondeur basé sur la vision. La méthodologie proposée utilise l'algorithme photos de l'historique des mouvements pour déterminer le mouvement d'une personne à travers un ensemble d'images dans un clip vidéo (RVB). À l'aide d'un réseau de neurones convolutifs , Nous avons créé un modèle de classification d'images pour détecter les chutes et les non-chutes. Le but des travaux est de détecter correctement les chutes et de réduire le nombre de fausses alarmes pour le système. Alors que le taux de précision de notre modèle a atteint 95,61 %.

Mots clés : Détection de chutes, Classification, Apprentissage approfondu, basé sur la vision.

ملخص

يعتبر السقوط من أكبر التهديدات التي يتعرض لها كبار السن وله تداعيات نفسية وجسدية ومالية كبيرة. إنه السبب الرئيسي للإصابات الخطيرة والإعاقات والاستشفاء وحتى الوفاة ، خاصة إذا لم يتلقوا المساعدة الطبية الطارئة. يؤثر وقت الاستجابة بشكل كبير على فعالية التدخل الطبي ويقلل من الضرر الذي يلحق بالمصابين. من أجل مساعدة كبار السن على ممارسة حياة طبيعية في المنزل ، تم اقتراح أنظمة مراقبة ذكية تكتشف السقوط وتنبه مقدمي الرعاية الصحية. في هذا الموضوع ، نقدم مثالاً لهذا النظام الذي يحاول حل مشكلة الكشف عن السقوط من خلال تقنيات التعلم العميق القائمة على الرؤية. تستخدم المنهجية المقترحة خوارزمية صور تاريخ الحركة لتحديد حركة الشخص من خلال مجموعة من الاطارات في مقطع فيديو. باستخدام شبكة عصبية تلافيفية، قمنا بانشاء نموذج للتصنيف الثنائي للصور لاكتشاف حالة السقوط وعدم السقوط. الغرض من هذا العمل هو اكتشاف السقوط بشكل صحيح وتقليل عدد الإنذارات الكاذبة للنظام. حيث بلغت نسبة دقة نموذجنا 95.61% .

الكلمات المفتاحية : الكشف عن السقوط ، تصنيف الصور، التعلم العميق ، القائمة على الرؤية.

Contents

List of Figures

List of Tables

List of Acronyms

FD	Fall Detection
FDD	Fall Detection Datasets
CNN	Convolution Neural Network
MLP	Multi-Layer Perception
DNN	Deep Neural Network
RNN	Reccurent Neural Network
LSTM	Long Short-Term Memory
WHO	World Health Organization
MFS	Morse Fall Scale
ML	Machine Learning
MHI	Motion History Image
MD	Motion history image Dataset
FPS	Frames Per Second
RGB	Red, Green, and Blue
MDtest	Test data in MD dataset
URFD	University of Rzeszow Fall Dataset.
MCFD	Multiple Cameras Fall Dataset

Chapter 1

General Introduction

1 Chapter 1 : General Introduction

1.1 Introduction

Falls are a major health concern worldwide due to severe short- and long-term consequences such as higher rates of paralysis, mortality, reduced functional ability, and prolonged admission to recovery facilities [Rubenstein LZ , 2006] . As we age, our bodies go through many physical changes to become more fragile and prone to falls [Lenise A , 2011]. For example ,our reaction times are slower, which is associated with an increased risk of falls ,Our vision is weaker which makes it difficult for us to identify obstacles and Medicines that make us sleepy or dizzy can impact our balance.
According to the World Health Organization [WHO ,2008] 30% of older persons fall at least once a year, and the frequency of falls rises exponentially with age. Meanwhile, the costs of caring for injured people have risen dramatically. As a result, the healthcare industry is interested in an automated system that can aid the elderly and staff in the case of a fall.

1.2 Background

To identify falls in the elderly, various approaches have been tried. Wearable device-based approaches, environmental sensor-based methods, and image-based methods are the three types we've identified. For wearable-based methods, most use 3-axis accelerometers to detect sudden changes in acceleration caused by a fall. However, because the elderly are prone to forgetfulness, wearable technologies are not ideal for them. Environmental sensor-based approaches, on the other hand, make use of a variety of sensors to detect and interpret falls in the environment. For instance, to detect falls, a smart floor with pressure-sensitive fiber sensors can be used. However, large-scale deployment of the sensors is both costly and unfeasible. The successful use of machine learning algorithms in picture recognition has recently boosted the popularity of vision-based technologies. Due to the limitations of wearable devices and ambient systems, vision-based systems provide feasible fall detection solutions. Basically, different systems based on different algorithms have been proposed, In general, several vision-based systems based on various algorithms have been presented, but they all share a basic standard architecture.

1.3 Motivation

Nothing beats working on research that has a huge impact on people's lives. It is important to protect the right of older people to lead their normal lives at home. However, studies have revealed that falls are a serious health problem that has a significant negative impact on the lives of older adults, making their safety a priority. Each year, 2.8 million older adults are treated in emergency departments for fall injuries. In 20% of falls, severe head injuries and bone fractures occur. Fall injuries cost $31 billion in direct medical costs, adjusted for inflation, the figures could also increase dramatically [H.Alkittawi, 2017] . Therefore, fall detection systems have become a necessity. Moreover, the rapid developments in the field of deep learning, as well as its accuracy in object recognition and image classification, have led us to believe that it could be a useful option in health monitoring of the elderly, especially in public places where surveillance cameras are common. Because vision-based fall detection systems may be combined with an existing monitoring system, they are much less expensive than sensor-based or wearable systems.

1.4 Objectives definition

The goal of this project is to develop a fall detection system that could be used as a subsystem to assist the elderly, support staff, and other individuals. The following features would have to be included in such a system:

- The system should work in real time, which means it should detect a fall as soon as it occurs. The fall detection algorithm, in particular, must operate at a faster rate than the input signal, which is the video stream.
- The system should be able to detect a fall from up to six meters away, which is reasonable from the perspective of a resident, given that a typical room is around 24 square meters, or 6x4 meters.
- Low intrusiveness is preferred because users should not feel that their rights were being violated.
- Interoperability ,means that the system must be accurate enough in detecting falls to allow other systems, such as an alarm system, to make decisions based on the system's predictions.

1.5 Thesis Organization

The remainder of this thesis is organized as follows:

- In the chapter 2 ,we will give literary reviews on the definition of falls, its kinds, the systems employed in the detection of falls, and some current works in the field of fall detection.
- In the chapter 3 ,we examine the fundamentals of machine learning, as well as the fundamental concepts of neural networks and deep learning, and explain some of the algorithms.
- In the chapter 4 ,the implementation of the system is described in more detail. Where we describe the system's overall structure, as well as the dataset that was used and other details.
- In the chapter 5 ,we evaluate the system in relation to the identification of the problem. We compare our solution to solutions that already exist, followed by an explanation of how it works and the disadvantages of the existing system.
- In the chapter 6, we concludes the thesis with delimitation and future vision.

Chapter 2

Litterature Review

2 Chapter 2 : Litterature Review

2.1 Introduction

Falls are a major public health problem for the elderly. Needless to say, the injuries caused by falls that elderly people experience have many consequences to their families, but also to the healthcare systems and to the society at large.that fall detection has drawn increasing attention from both academia and industry, especially in last years, where a sudden increase can be observed. Moreover, on the same line, the topic of fall-likelihood prediction is very significant too, which is coupled with some applications focused on prevention and protection. Essentially, to clearly understand and undertake the research problem, an empirical framework ought to be outlined to categorically outline a fall and highlight challenges experienced after a fall. Furthermore, causes of falls in elderly patients' aid in conceptualizing measures ought to be undertaken for fall detection. Moreover, a proper review of the implemented technologies that aid in detecting falls by patients provides a framework to undertake the study. Notably, different technologies have been proposed to aid in detecting falls each with a number of advantages and challenges. A conceptual framework provides an outline of the study to be undertaken.

2.2 Falls

Falls are divided into two categories: fatal and non-fatal. Every year, around 684,000 people die as a result of falls, demonstrating their seriousness [WHO, 2021] . It should be highlighted that falling is the most common cause of subsequent difficulties. Broken hips and traumatic brain injuries are common among the elderly. Trauma accidents primarily affect older persons' lives since they are three times more prone to fall following a fall. Fall accidents varies in terms of their forms and severity, as well as their causes. The study and classification of these accidents enables the elderly and those who care for them to take the necessary preventative measures.

2.2.1 Definition

A fall is commonly visualized rather than defined . A fall, according to the World Health Organization [WHO, 2021] , occurs when a person comes to rest accidently on the ground , or other lower level. Falls can cause fatal or non-fatal injuries, such as broken bones or bruises. Many people have no signs before a fall, but some do experience dizziness or other symptoms.

2.2.2 Types of Falls

Patients fall for a variety of reasons, and if falls are to be prevented, it is critical to understand the etiology of a fall. Analysis of circumstances surrounding 100 patients who fell and 100 randomly selected patients who had not fallen [Morse et al, 1987] revealed that three types of patient falls occurred in hospitals and long-term care institutions. Because falls have different causes, the strategies for preventing patient falls are different for each type of fall. A fall may be classified as accidental or physiological, with the physiological falls further classified as predictable that is, an anticipated physiological fall (i.e. the patient exhibits signs that indicates the likelihood of falling and scores at risk on the " Morse Fall Scale " MFS) or as unpredictable that is, an unanticipated physiological fall.

a **-Accidental Falls**

Fourteen percent of all falls are accidental, caused by a patient tripping or slipping. The causes are mostly due to environmental factors, for example unavailability of grab rails, slippery floors due to spilled water and other factors, insufficient lighting, bed and toilets of improper height, obstructive corridors, and carpeting. Inverted edges and raised door sills. It can also occur because the patient makes errors in judgment, such as leaning on unstable objects or when moving from bed to a wheelchair. Accidental falls cannot be predicted using any scale like other types because it is not due to physical factors but is due to environmental hazards or miscalculation, prevention strategies are designed to ensure that the environment is free from risks, direct the patient towards the environment, and receive instructions on how to do so . For the use of walkers, etc. This includes instructions on the correct way to get out of a wheelchair. [Janice Morse, 2008]

b **-Anticipated Physiological Falls**

These are falls that occur in patients identified as being at risk with the Morse Fall Scale (MFS) assessment test, which is a quick and simple way to assess a patient's likelihood of falling. It represents six factors that significantly contribute to a patient's likelihood of falling, namely (history of the fall, secondary diagnosis,

ambulatory aid, intravenous therapy, type of gait and mental state) . Anticipated physiological falls constitute 78% of all falls. [Janice Morse, 2008].

c **-UNAnticipated Physiological Falls**

Unexpected physiological fall these are falls that may be attributable to physiological causes, but arise from unpredictable conditions before they first occur. They make up approximately 8% of all waterfalls. Examples include seizures, 'falling attacks', syncope, or pathological fracture of the hip. Depending on the cause, when this type of fall occurs, nursing attention is directed either toward preventing a second fall or preventing injury when the patient falls again, as there is a possibility that the underlying condition will recur. For example, we may teach a patient with orthostatic hypotension how to recognize dizziness when rising, and how to get up slowly, thus reducing the risk of falls.[Janice Morse, 2008].

2.2.3 Importance of Fall Detection in the Society

a **- Healthcare Costs**

Falls in the elderly is a major public health problem as they can lead to irreversible health ,social ,and psychological consequences, and a large economic burden. More than one-third of persons 65 years of age and older fall each year ,and a half of falls are recurrent .
Falls are not only a risk factor for fractures but also for development of traumatic cerebral or visceral hemorrhagia ,traumatic pain syndromes, functional limitations, dislocations ,soft tissue injuries, excess healthcare costs ,and increased mortality .
In the year 2010 in the USA, there was an estimated 10,300 fatal and 2.6 million nonfatal but medically treated fall-related injuries in individuals above the age of 65 years. Of the fall-related injuries identified through the surveillance system in Florida, about 42% resulted in hospital admission and about 50% of fall injury events that occurred at home and required hospital admission resulted in a person being discharged to a nursing home . [Medicina, 2015]. As the population of elderly people is growing fall-related injuries affect a substantial number of older adults. It is likely that the overall numbers and costs of fall-related injury hospitalizations will continue to rise .
Several studies have been performed in different countries estimating the economic burden of falls . It is difficult to compare the costs between countries due to differences in populations investigated ,study methods, and expenses of medical care ,but the magnitude of economic burden was demonstrated to be significant in all these publications.

b **- Prevalent Injuries**

A fall can change your day, your week, and maybe even your life. These injuries can often happen anywhere and to anyone, which can be frightening in their suddenness ,and cause severe pain and injury. Falls are the most common variety of accidental injuries. However ,a fall is not a type of injury ,it is a cause.
Sometimes people fall ,and don't feel any significant pain until the next day ,but when they do begin to feel pain ,it is often associated with discomfort and limited range of motion. Those symptoms might mean something more serious.
Additionally, other fractures such as femur, pelvis, leg and ankle fractures result from falls. However, hip fractures account for most of the health problems in the elderly from falls since they affect their mobility. Furthermore, according to [Lídia Sànchez-Riera et al, 2017] mortality rate increase was closely associated with an occurrence of a hip fracture.
As such, hip fractures affected the lives of elderly patients with increased dependence, decreased social engagements and reduced quality of life. Intracranial Injury leads to traumatic brain injury. Notably, falls result in over one-thirds of reported TBI in the population in the United States.
Among the elderly patients, falls lead up to 4 out of 5 cases of Traumatic brain injury "TBI". As such, falls results in predominant injuries to the elderly patients.

c **- Quality of Life**

Women with wrist fractures had higher quality of life declines than men 6 months after the fall in all questionnaires except the LBI, and greater declines than the non-fracture sample regardless of gender. Following wrist and hip fractures, patients over the age of 80 had lower baseline scores on all end measures and lost more HRQoL (Health-related quality of life) and functionality. Patients' quality of life continues to decrease after a fall. Fifty percent of all injury-related issues that prohibit the elderly from doing daily duties are

caused by falls. [Taguchi et al, 2016] discovered that elderly people who have a high quality of life have fewer falls. Furthermore, those who have undergone treatment for fall-related injuries, such as hip fractures, lose their capacity to do daily tasks. Specific prophylactic actions may be required to minimize or lessen the impact of fractures caused by falls. Many seniors will benefit from preventive tactics because of the residue it leaves on the elderly who fall, who are unable to resume their activity and movements previous to the accident and may gradually lose their ability to walk.

2.2.4 Fall Detection Architecture

In all fall detection systems the overall system structure of the fall detection system is similar. As the system consists of different components that help in the fall detection process.
The model architecture for the implemented systems comprises of similar architecture. The figure (1) depicts the basic architecture for fall detection systems:

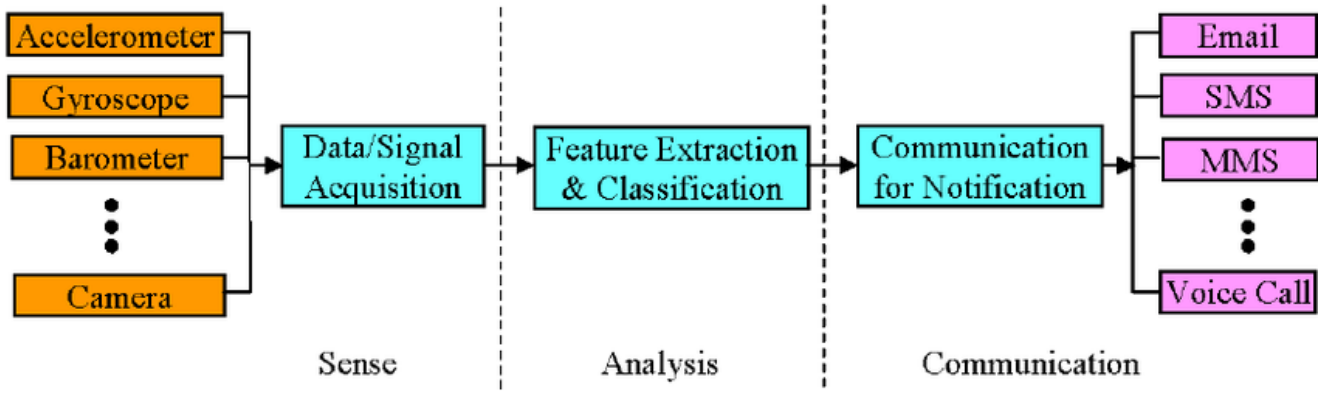

Figure 1: Fall Detection Architecture [Habib, 2014]

a **- Sense**

This is the first phase of any fall detection and prevention system, and it involves sensing or measuring acceptable physical quantities with appropriate sensors. This component is critical in supplying the necessary data input to the system in order for it to identify a fall. This component, in particular, consists of several sensors that can be utilized to retrieve data. The sensors could include an accelerometer, gyroscope, barometer, and camera, as shown in Figure figure (1). This is significant since it is used to classify fall detection systems. Wearable-device-based systems, ambient-based systems, and vision-based systems are all examples of this.

b **- Analysis**

The obtained signals/data should be analyzed after the physical quantities are measured using sensors. The relevant features from the sensor's outputs are retrieved in this step, and preliminary choices are made by classifying and analyzing the extracted features.
Threshold-based algorithms and Machine Learning-based algorithms are the two types of algorithms used by fall detection systems. To make early judgments concerning a possible fall event Threshold-based algorithms match the sensor's outputs to a predetermined threshold value (s). Threshold-based algorithms may employ several thresholds, with the threshold value(s) being either fixed or adaptive. In order to detect falls, the system uses pre-programmed values entered by the system engineers.
Machine-learning based systems, on the other hand, use machine learning techniques to identify a fall. The algorithm, in essence, classifies an event as a fall or non-fall based on accurate feature extraction from sensor input. Hidden Markov Models and Convolution Neural Networks are among the machine learning algorithms used in these systems.

c **- Communication**

Communication Algorithms in the second phase provide preliminary identification or prediction of falls events based on sensor responses from the first phase. When the system detects or predicts a fall incident, it notifies the user and/or carers.

This communication phase is usually split into two parts in most fall detection technologies. The system tries to get feedback from the user by checking the preliminary choice in the first stage, which improves the system's sensitivity. The user's reaction determines the next stage. The system resets if the user actively rejects the suspected fall. Otherwise, carers will receive a notice requesting quick assistance. Some systems do not wait for user feedback and send an alarm message to the caregiver right away. Fall prevention systems often inform users about their impending fall rather than soliciting comments.
In addition, rather than notifying users, fall prevention systems can also trigger additional assistive systems (e.g., wearable airbag, intelligent walker, intelligent cane, intelligent shoe, etc.) to protect the user from the negative impacts of falling. By analyzing the sensor's output, user input can be collected automatically. [Sposaro and Tyson's, 2009] method, for example, derives the final decision by automatically assessing the difference in location data before and after the alleged fall event. Other systems require the user to provide manual feedback. The user's feedback is also collected via a combination of alert systems. The system waits for a pre-defined period (usually 1 minute) after requesting a response from the user. If the user does not answer within that time frame, the system will treat the situation as a fall. Automatic fall detection systems may fail to detect a true fall event. In such instances, some systems feature help (or panic) buttons, allowing users to manually seek assistance.
Different ways of communication have been employed to deliver the message from the fall detection system to interested parties, as shown in Figure 1. The primary notification modes for fall detection systems are email, short messages, multimedia messaging, and voice calls. Notification messages may include information such as time, GPS location (coordinates), and a geographical map.

2.2.5 Fall detection systems

Technology plays a significant part in the evolution of the healthcare industry. Most importantly, the employment of assis- tive or adaptive technology in the medical profession helps the patients to acquire a degree of independence. However, older patients still require proper surveillance to warn medical practitioners in the case of emergency. As such, many technologies have been developed to aid with fall detection for the elderly. Notably, the categories of the systems depend on the technology utilized to give the answer. According to [Kaur, 2017], fall detection systems can be divided into wearable systems, vision-based systems and ambiance sensor-based systems. Furthermore, each of the systems utilizes distinct approaches to detect falls. Wearable systems may be split into inactivity based or posture-based systems. Additionally, vision-based sys- tems fall in categories : Body shape change analysis, inactivity detection, and 3D head changeanalysis [Nizam et al, 2016]. Lastly, ambience-based systems fall in two groups : Floor sensor detectors and posture. Essentially, each of the offered systems presents a different way to detecting falls.

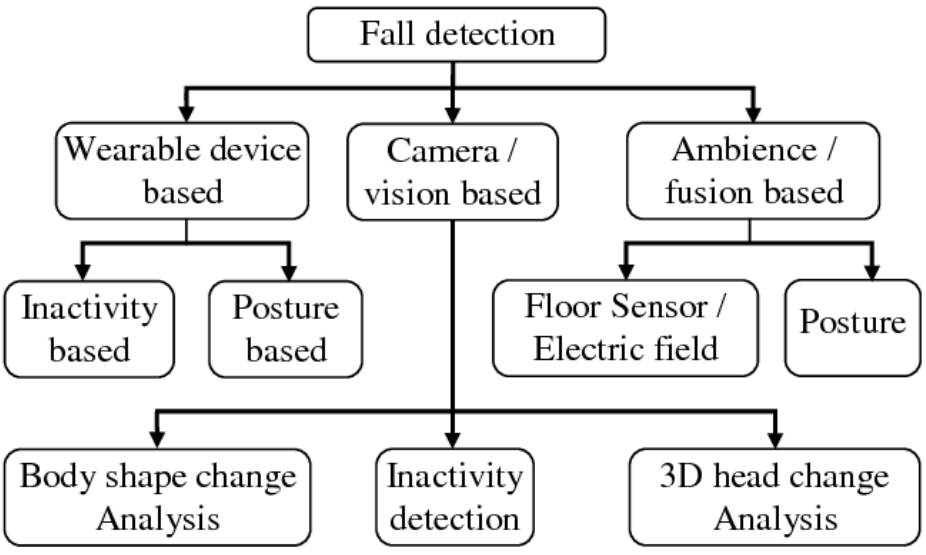

Figure 2: Fall Detection Classification [Nizam, 2016]

- **Wearable Device Based Systems**

  These systems detect falls as they are attached to the patient's body or their clothes. Sensors attached to the body determine the posture of the patient and the acceleration. As such, this determines whether the patient has fallen or is falling. Essentially, accelerometers determine the acceleration of the body while the gyroscopes determine the change in angular velocity of the body. Notably, this highlights the difference in technologies used in the wearable device based systems.

  Furthermore, the implementation of the technologies on fall detection systems differ as similar devices may be utilized with different approaches. Noteworthy, apart from the different technologies applied in fall detection, there exist different algorithms applied that aim at fall detection.
  As such, the precision of a fall detection system depends on the recognition features chosen in the system. For instance, in wearable system using accelerometers, the change in acceleration maybe utilized as the recognition feature. Essentially, with the recognition features, the detection algorithms can be categorized as either threshold-based or machine learning based systems.
  Threshold based systems rely on pre-defined parameters referred to as the thresholds outlined by the system engineers thus lead to a fast system response time with reduced resource consumption. However, precise predefined parameters ought to be provided to improve the accuracy of the system.
  On the other hand, **machine learning based systems** utilize classification of the parameters to aid in fall detection. Notably, this leads to system overhead as the approach utilizes more computing resources. Therefore, due to the limited resource on wearable devices then most of the devices utilize threshold based systems.

  Accelerometers aid in fall detection through a threshold-based system comparing an object's acceleration. Essentially, a tri-axial accelerometer provides a body's acceleration in three different orientations that is the X, Y, and Z. As such, this may be evaluated to aid in detecting a fall.
  Mostly, the systems detect a fall if there is an abrupt change in the body's orientation resulting in a lying position coupled with a prior negative acceleration. [Wu et al, 2015] proposed a wearable device that uses an accelerometer to detect an elderly patient falling through an analysis of the acceleration.The total acceleration magnitude is calculated as shown in equation(1).

$$|a| = \sqrt{x^2} + y^2 + z^2 \tag{1}$$

  Notably, in a fall scenario, due to the impact with the ground or surface, there is a high value recorded in the sum of the acceleration. Further, the system asserts a fall decision by calculating the change in angle that is calculated as a result of the acceleration values. Since the system utilizes a threshold approach, if the calculated values surpass the required thresholds then a fall is detected. The system then identifies the geographic position of the person and sends a short alarm message to the care providers.
  Wearable device based systems present a non-invasive and privacy compliant method to fall detection. However, the systems fail in accurately detecting a fall from an activity of daily living. Moreover, the systems have high power consumption due to the sensors. Lastly, elderly patients may unintentionally not wear the device therefore presenting a challenge on the use of the technology.

- **Ambience Device Based Systems**

  Ambient device based systems detect falls with the use of sensors that measure the sound, pressure, vibration and motion of an individual on a surface. As such, the systems utilize the environment to enable fall detection. Systems under this category use pressure and vibration sensors to aid in distinguishing normal daily activities and falls. As such, they consider that pressure on the surface caused from a fall is different from pressure exerted by a normal daily activity. [Alwan et al, 2006] proposed a system that detects floor vibrations and classifies them as falls or non-falls. If a fall is detected the system then sends a notification to the relevant authorities. As highlighted earlier, the system assumes that vibration signature generated from a fall is different from the signature generated from a normal activity.
  The system computes the vibration pattern with features such as the frequency, amplitude, duration and succession. The evaluation of the features and the values aid in identifying a fall. Notably, the system suffers from the challenge of false alarms. Essentially, different surface materials lead to different vibration patterns. Further, the weight of an object influences the vibration pattern of the object.

  [Sixsmith and Johnson, 2004] Sixsmith and Johnson (2004) presented a system that detects falls with the aid of a pyro electric infrared sensors. The sensors monitor the movement of an object on a surface thereby

detecting a fall. Essentially, they provide the velocity, location and size of the object which is key in determining a fall.
As such, this reduces the number of false positives that may be presented with the use of only the sensors. The proposal leveraged on the use of readily available and cheap sensors. However, the applicability of the system on a real environment presents issues due to the power consumption of the sensors.
Ambient-Based systems present a novel approach to fall detection. However, the systems suffer from the challenge of false positives. Mostly, this is due to the fact that the sensors may pick noise as they aggregate the data. Furthermore, the systems rely on sensors faced with the challenge of their power consumption.

- **Vision-Based Systems**

Lately, vision based systems have gained popularity due to the successful implementation of machine learning algorithms in image recognition. Over the years, wearable systems and ambient systems have played a major role in fall detection. Notably, most of the implemented systems use wearable devices.
However, due to the limitations of the wearable devices and ambient systems, vision based systems provide feasible solutions to fall detection. Essentially, different systems based on different algorithms have been proposed, however they have a standard architecture as discussed below:

1. **Vision-Based Systems Architecture**

   a **- Camera**
   The camera serves as the primary sensor for vision based systems. Notably, the systems provide a real-time analysis of a fall algorithm to detect a fall. As such, the need to use fair priced cameras and resourceful computing platforms. Currently developed systems use 2D vision cameras that provide images of the objects. However, some of the systems have utilized 3D time of flight cameras that provide a 3D illumination of the object.

   b **- Processing Unit**
   Vision based systems require high computing resources for the images. Essentially, for a real-time fall detection system, an analysis of one or a series of images requires fast processing units. Therefore, the need for a fast processing unit.

   c **- Background Subtraction**
   The captured images contain data that is not relevant to the data analysis. Therefore, to focus on an individual on an image, background subtraction ensures that not relevant information on the image is removed. This reduces the required processing of the image in fall detection. Furthermore, this ensures that the fall detection system utilizes the correct image of a person. This can be accomplished with the use of an estimation model that aids in background subtraction. The models are classified as recursive models and non-recursive models.
   Non-recursive models store a buffer memory with a series of previous frames that aids in the computation of the background model. Therefore, the approach uses a time sliding window function that aids in calculation of the required model. The developed models adapt to frequent changes since the model generated is based on the previous number of buffered frames not all the frames. However, for the implementation of the system, the system requires adequate buffer memory to store the images. Notable techniques in this model include: Median filtering and linear predictive filter.
   Recursive models update the background model with new occurrence of a new frame. Therefore, the model stores only one image in the memory as such reducing the storage memory. However, the occurrence of a problem in one of the frame leads to the subsequent issues in the background image. As such, the model does not adapt to new changes in frames such as the non-recursive model. The techniques implemented under this method include: running average, Kalman filtering, approximated median filtering and Gaussian mixtures.

2. **Vision-Based System Classifications**
   Vision based systems can be categorized into three based on the mode of operation:

a **- Inactivity**

The systems assume that a person is inactive following a fall. As a result, the systems keep an eye on a person to detect idleness. [Jansen et Deklerck, 2006] suggested a technique that uses a human silhouette to capture a person's 3D orientation. The system then looks at a series of photos for inactivity. It includes a context-dependent inactivity detection feature that obtains the inactivity's location, time, and duration. This aims to lower the number of false positives.

The system may, for example, recognize that a subject is laying on the bed without issuing an alert. Evidently, it employs a model that distinguishes between typical everyday activities and the occurrence of falls. As a result, the model may adapt to different tasks. Because falls occur at random, the system does not totally present a good approach to fall detection. As a result, using the context dependent inactivity method in real-life random falls poses a hurdle. There was also no validation of the system in a situation other than the simulated experiment, as the researchers point out.

[Belshaw et al, 2011] suggested a system that combines body shape analysis and detection of inactivity. Images of patients uploaded to the system are evaluated in this manner to detect a fall, but some portions in a room have been divided as inactive sections. A person on a bed, for example, may not be defined as falling because the region around the bed symbolizes an area of inactivity.

The researchers generate a silhouette of the subject, extract attributes, and classify the state as a fall or non-fall for fall detection based on photographs. The method obtains a true positive rate of 97 percent and a false positive rate of 5% using neural network classifiers.

b **- Head Motion Analysis**

The Head Motion Analysis approach to fall detection is based on the concept that the rate of change of vertical motion is higher than horizontal motion during a fall. To identify a fall, the technology analyzes the rate of change of an individual's head.

[Rougier et al, 2006] suggested a three-dimensional head tracking system for detecting falls. Essentially, the method implies that an individual's head is visible in an image. The head's trajectory can thus be calculated using a particle filter. As a result, by measuring the motion of the head, the threshold velocity for fall detection can be determined, allowing a fall to be identified.

The approach has a number of drawbacks, including improper head recognition, which can lead to incorrect fall detection. Furthermore, there has been less research on 3D head motion analysis, which has an impact on the approach's accuracy.

c **- Shape Change Analysis**

The method posits that a human's body form changes over time during a fall occurrence. As a result, this method tries to detect changes in the structure of the human body in order to detect a fall or a daily activity. Machine-learning and threshold-based fall detection algorithms, as previously mentioned, play a vital role in accurate detection.

[Rougier et al, 2007] proposed a shape change analysis method based on thresholds. The method uses Motion History Image to assess an individual's change in motion, which is combined with an examination of the person's shape to determine a fall. Changes in an individual's movements in a video sequence can be identified using optical flow. With the usage of a real-time fall detection system, this poses a problem.

As a result, the employment of a Motion History Image, which gives a static representation of an object's change in motion. A silhouette's pixel intensity indicates an object's most recent position or motion in a video. The figures (3) below depict (a) an individual walking, (b) an individual falling, and (c) an individual going about their daily activities. This is noteworthy since it demonstrates an accurate method for detecting inactivity.

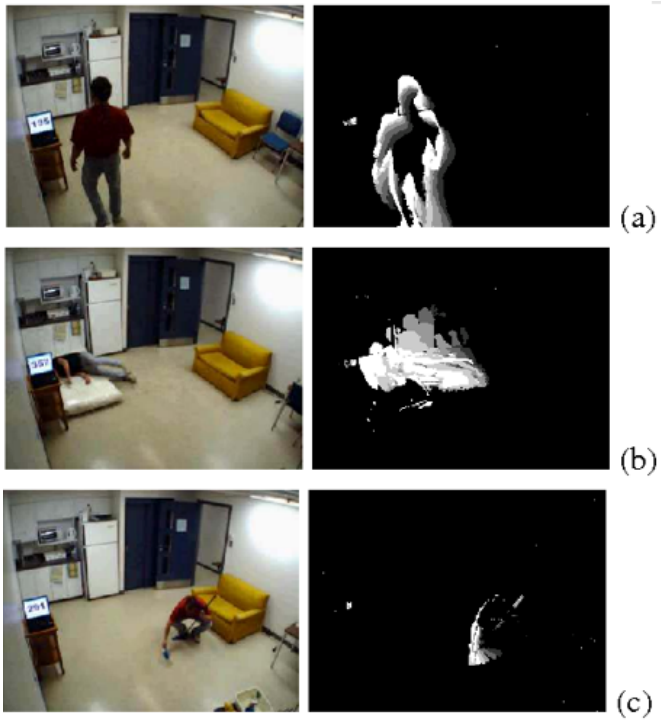

Figure 3: Motion History Image

The collected features are then loaded into a Hidden Markov Model (HMM) that distinguishes between a fall and a non-fall. On less complex fall detection scenarios, the system achieves a level of accuracy of 100%, and on challenging fall detection scenarios, it achieves a low of 82.35%.
The algorithm's average performance, on the other hand, is 87.38%. The system, in particular, suffers a time constraint in processing video analysis of the algorithm to detect a fall. This effectively increases the time it takes to detect and alarm in the event of a fall from 1 to 5 seconds. This is clearly due to the calculation processes used in each frame, such as background subtraction and feature extraction.

A convolution neural network fall detection system was proposed by [Nez-Marcos et al, 2017]. Optical flow images are used as input to the network to help with feature extraction for fall detection. Image processing processes such as background reduction are avoided to improve the system's performance and reduce processing time.
Previously, image processing was blamed for the lengthy processing time of fall detection systems. The adaptability of the CNN to be trained on diverse datasets is applied in the system to boost the algorithm's robustness and accuracy due to the restricted amount of datasets in fall detection. Various datasets are used to train the model with various picture properties that are significant in fall detection. Furthermore, optical flow images assist the network in identifying an individual's various actions. Finally, fall detection is addressed through the application of transfer-learning and network layer tweaking.

3. **Object Detection Algorithms**
The techniques discussed above serve as the foundation for detecting a fall in an image. The methods are particularly useful when there is only one primary item (a person) in an image. As a result, the methods described above rely on the presence of one object in each of the photos under consideration. In a real-world setting, numerous people can appear in a single image frame, challenging the usage of the vision-based techniques described above. Object detection is the process of detecting items in an image and creating a bounding box around them. Different object detection algorithms have been developed with time which based on the Convolution Neural Network Architecture:

a **- Region-Based Convolution Neural Network**
[Girshick et al, 2014] introduced the Region-Based Convolution Neural Network (R-CNN) to overcome object detection region proposal difficulties. The architecture suggests using a method that extracts 2000 zones from an image using a selective search algorithm.
The image's selected sections are twisted into squares and put into a convolution neural network, which produces a feature vector. To classify the presence of an object within the region specified, the retrieved feature vector is sent to a Support Vector Machine. Additionally, the technique generates offset values that aid in the process of changing the bounding box to improve the precision of the region proposal.
The figure (4) shows the architecture of the algorithm:

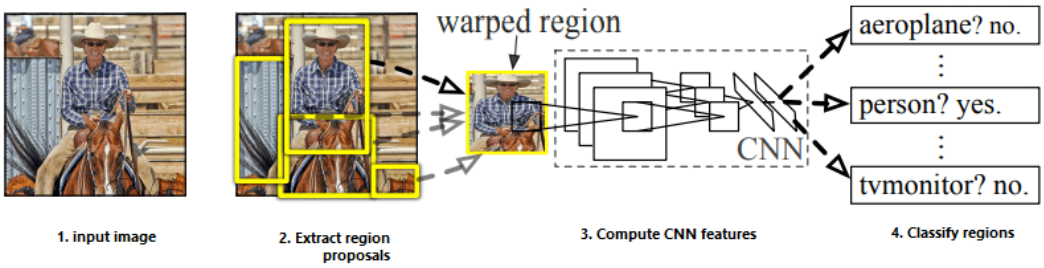

Figure 4: R-CNN [Girshick,2014]

The input image recorded by a camera is shown in the first image. The following image depicts the several proposed regions, which total 2000. Furthermore, the following graphic depicts the use of Convolution Neural Networks to compute or derive required features from the proposed regions. Finally, the network determines whether the region belongs to a specific class.

b **- Fast Region-Based Convolution Neural Network**
[Girshick et al, 2015] suggested the Fast Region-Based Convolution Neural Network (Fast R-CNN) as an improvement to the implemented Region Convolution Neural Network. An image is fed into the network, along with object suggestions. This allows the network to process the image using convolution neural networks and max pooling layers, and then produce convolution feature maps. The convolution feature map allows the network to recognize area proposals that are twisted into squares, and the proposal regions are reshaped using a Region of Interest (RoI) pooling layer before being supplied to a fully connected layer. A softmax layer is created from the RoI feature vector to forecast the specified class. the figure (5) shows the architecture of Fast RCNN:

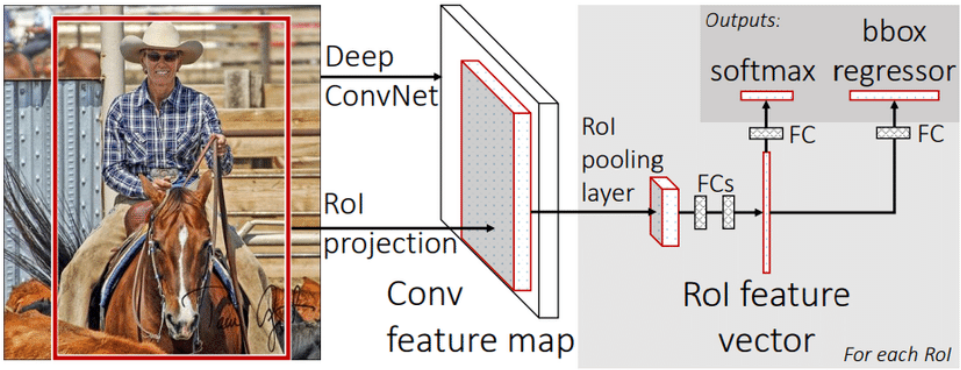

Figure 5: Fast RCNN [Girshick,2015]

The network's performance improved once the convolution operation was used to replace the process of producing region recommendations. Essentially, this reduces the amount of time it takes

to process each image during training and testing. Though the network is faster at processing images, the adoption of region suggestions has a substantial impact on the network's performance time.

c **- Mask Region-Based Convolution Neural Network**
Mask R-CNN extends Faster R-implementation CNN's to construct object masks for each of the image's detected objects. Proponents of the Faster RCNN model suggested the algorithm [Girshick, et al, 2017] This allows the model to provide accurate object instance segmentation for each object in the image. Essentially, the process to solve the problem entails to main steps:

a. Undertake object detection by drawing a bounding box in each instance of a class.
b. Undertake semantic segmentation on each of the identified bounding boxes.

Notably, if the system achieves high levels of accuracy in the first stage, it will also achieve high levels of accuracy in the second step. As previously stated, the model makes use of Faster R-capabilities CNN's to accurately perform semantic segmentation.
As a result, the algorithm adds a segmentation masks branch to each Region of Interest (RoI) that runs alongside the classification branch. This allows the algorithm to quickly construct pixel-wise masks for the various instances of each class. As a result of its capabilities, the algorithm will be applied to aid in fall detection in a multi-person scenario.

2.3 Previous works

In previous works, several methods were used to detect falls in the elderly. We classify them into three categories: wearable-based, environmental-sensor-based, and image-based methods. We will describe each of them in the following paragraphs :

a **- Wearable-Device-Based Methods**
[Rucco et al, 2018] offered a review of wearable device-based approaches to fall evaluation, prevention, and detection. According to their survey, triaxial accelerometers were used to capture fall signals in the majority of wearable-device-based techniques. Aside from that, the analysis reveals that the most commonly employed body section for wearable gadget sensors is the trunk, which is closely linked to walking.

To capture acceleration readings, [Lai et al, 2011] used triaxial accelerometers scattered throughout the body. The system found that the acceleration surpassed a pre-set threshold, indicating a fall accident. By comparing the acceleration at impact with normal acceleration, their method might also estimate the severity of the injury.

Similarly, [Pannurat et al, 2017] created a hybrid system for detecting distinct phases of falls that combines activity classification using machine learning and a rule-based knowledge representation. It can also be used for the head, arm, wrist, ankle, chest, side waist, front waist, and thigh. For all of the locations, as well as the fact that changing the wearable position does not require a calibration step.

[Wang et al, 2014] The use of accelerometer and barometer data to create a low-power fall detector was proposed. The use of the sensor on the user's chest was considered. The usage of the barometer sensor in their work could explain our algorithm's poor performance using most of the sensitivity settings when compared to their work.

[Pierlenoi et al, 2016] and [Sabatini et al, 2016] employed gyroscope, magnetometer, and barometer sensors in addition to triaxial accelerometers to recognize user posture.

[Aziz et al, 2011] Three accelerometers were put at the sternum, right ankle, and left ankle, and a linear discriminant analysis algorithm was used to determine three causes of falling.

[Bagalà et al, 2012] used a lower back accelerometer. Algorithm With posture recognition, a comparison of many threshold-based algorithms is made.

Location sensors were utilized by [Lustrek et al, 2015] to locate the user. The system raised an alarm when the user fell to the ground.

However, the elderly usually forget to wear wearable devices and smartphones. Thus, the existing wearable-device-based methods are not practical for long-term use.

b **- Environmental-Sensor-Based Methods**

Environmental sensors such as pressure, sound, and radar sensors have been used in several studies to detect falls.

[Feng et al, 2016] employed a smart floor with pressure-sensitive fiber sensors to detect falls by analyzing feature-specific pressure images comprising motion information. The device could be installed anywhere, even the restroom.

[Daher et al, 2017] employed force sensors and accelerometers hidden behind intelligent tiles to detect, track, and recognize user movements like walking, standing, sitting, lying down, and falling. The proposed approach, however, was expensive and could only be used indoors.

[Li et al, 2012] presented the acoustic-FADE approach for detecting falls using sound sensors. The system is made up of a circular microphone array that can pick up sounds inside. When a sound is detected, the system locates the source, amplifies the signal, and uses machine learning to classify the event as a "fall" or "non-fall." However, installing a circular microphone array in every room is impossible. Furthermore, when there are multiple people in the room, classification becomes more difficult.

[Su et al, 2015] used a ceiling-mounted Doppler range control radar to detect human falls. The radar detects motions from both falls and non-falls due to Doppler effects.

Similarly, [Shiba et al, 2017] presented a microwave Doppler sensor-based system. The system uses a hidden Markov model to calculate the trajectories of the frequency distributions that correspond to the velocities of the motions while falling. The amount of persons in the room, on the other hand, can readily impact the signal's analysis.

In a toilet room where the temperature should be less than 31 °C, [Kido et al, 2009] employed thermal imaging sensors to distinguish between normal and falling activities. However, due to privacy concerns, the proposed method's usefulness may be limited.

In summary, the accuracy of environmental-sensor-based methods can be easily affected by objects in the environment. In addition, pervasively deploying sensors is costly and impractical.

c **- Image-Based Methods**

Most of the image-based methods can be categorized into four types: multi-camera, single-camera, depth-camera, and wearable-camera methods.

1. **For multi-camera methods**
   [Auvinet et al, 2011] used numerous cameras to reconstruct 3D models of humans and identified falls by assessing volume distribution along the vertical axis. However, tracking people in public places with several cameras is difficult and impractical.

2. **For single-camera methods**
[Brulin et al, 2012] suggested a fuzzy logic-based posture identification algorithm. They used machine learning technologies to detect a person amid other moving objects.

[Yu et al, 2012] proposed a human silhouette-based posture identification system as well. The object with the most moving pixels is considered a human body when identifying a human body among moving objects. Then, to describe different postures, elliptical fitting and a projection histogram are utilized. Finally, the posture is classified using a directed acyclic graph support vector machine (DAGSVM). If the person is lying on the ground, however, the procedure may give a false alarm.

[Mirmahboub et al, 2013] found a human silhouette using two distinct background separation methods and used the silhouette's area as a feature to feed into a support vector machine (SVM) for classification.

Background subtraction was utilized by [Agrawal et al, 2017] to detect things in the foreground, and contour based human template matching was employed to categorize a human. They calculated the distance between the top and mid-center of a human's bounding box to identify a fall.

To detect human objects, [Poonsri et al, 2018] used a background removal and Gaussian mixture model. They then extracted features and classified the postures using the orientation, aspect ratio, and area ratio. Background subtraction, on the other hand, may not be able to distinguish several human objects effectively, resulting in an incorrect classification result.

Furthermore, if the person is obscured by other objects, the above mentioned methods may not work properly.

3. **For depth-camera methods**
[Ma et al, 2014] proposed an approach in which each frame's curvature scale space (CSS) properties are extracted and the action is represented by a bag of CSS (BoCSS) words. They then use the extreme learning machine (ELM) classifier to distinguish the BoCSS representation of a fall from those of other acts.
[Bian et al, 2015] proposed utilizing the SVM classifier with the input of the 3D trajectory of the head joint to detect the motion of a fall.

[Auvinet et al, 2011] used many cameras to create a three-dimensional (3D) shape of an elderly person, which they subsequently examined along the vertical axis.

To detect falls, [Diraco et al, 2010] measured the distance between a 3D human centroid and the floor plane.
[Angal et al, 2016] collected information on the velocity, acceleration, and width height ratio of a human item using the Microsoft Kinect sensor to detect a fall.

However, the above-mentioned methods may not be accurate when the person lies on the ground. In addition, these methods cannot recognize the difference between falls and sitting on the ground. In summary, all the above-mentioned methods focus on falls that occur during forward walking. They cannot accurately detect falls that occur while sitting down and standing up from a chair.

4. **For wearable-camera methods**
Wearable cameras collect data from users that can be used to assess an environment, recognize interactions, and categorize bodily activity. In addition to the imager, a wearable camera has several sensors such as microphones and inertial measurement units.

In contrast to previous vision-based systems, [Ozcan et al, 2016] took the opposite approach by taking a completely different perspective. The system used a modified version of the histograms of oriented gradients (HOG) technique in conjunction with gradient local binary patterns (GLBP) to identify falls

by comparing dissimilarity distances calculated by HOG and GLBP. However, because the elderly are prone to forgetfulness, wearable camera systems are not feasible for daily use.

However, despite this richness in sensing modalities, the analysis of data from a wearable camera is particularly challenging due to unconventional mounting and capturing conditions, rapid changes in camera pose, self-occlusions, background noise and motion blur.

2.4 Conclusion

In this chapter, we have provided a detailed explanation regarding the subject of falling detection, where we explained the types of falls and the techniques used to discover and deal with each type. We also presented an explanation of the systems that have been proposed and developed so far to detect falls. We also mentioned some previous work and experiences in this field. This will help us clearly understand and deal with the problems, and choose the appropriate approach to develop our own system for detecting falls in the elderly.

Chapter 3

Therotical Basis

3 Chapter 3 : Theoretical Basis

3.1 Introduction

First attempts to build analytical models relied on explicitly programming known relationships, procedures, and decision logic into intelligent systems through handcrafted rules (e.g., expert systems for medical diagnoses). Fueled by the practicability of new programming frameworks, data availability,and the broad access to necessary computing power, analytical models are nowadays increasingly built using what is generally referred to as ML ,machine learning seeks to automatically learn meaningful relationships and patterns from examples and observations. this relieves human of the burden to explicate and formalize his or her knowledge into a machine-accessible form and allows to develop intelligent systems more efficiently.
Deep learning is a type of machine learning that uses artificial neural networks to learn. Deep learning models outperform shallow machine learning models and traditional data analysis approaches in many applications. In this chapter, we review the fundamentals of machine learning and deep learning in order to have a better grasp of the methodology behind today's intelligent systems. We give a theoretical description of the methodologies and algorithms utilized to develop the automated analytic model using machine learning and deep learning, as well as a conceptual differentiation between related terminology and concepts.

3.2 Machine Learning concepts

Machine learning is a branch of artificial intelligence and a subject of study that attempts to make computers able to learn and improve on their own based on their past experiences. Machine learning focuses on the ability of computer programs to learn from a variety of observations. Machine learning is gaining popularity in a range of business areas and is rapidly increasing in the world of information technology. Despite the fact that machine learning has only gained great popularity in recent decades, it is widely used in all technologies. In this subject,we will cover some of the basic topics of machine learning, such as what machine learning is and the techniques and algorithms used in it.

3.2.1 Fundamentals of machine learning

Machine learning can be defined as a branch of artificial intelligence where we design computer models capable of learning from a given dataset with minimal human intervention. According to [Murphy , 2012] ,Machine learning is defined as a set of methods that can automatically detect patterns in data and then use the uncovered patterns to predict future data or perform other types of decision making under uncertainty. A machine learning model can be predictive if it forecasts future conditions, descriptive if its goal is to gain knowledge from given data, or both predictive and descriptive. We will highlight a number of fundamental machine learning concepts.
The fundamental learning process can be divided into two steps, a training phase and a testing phase, which require two different types of data sets, as depicted in Figure (6).

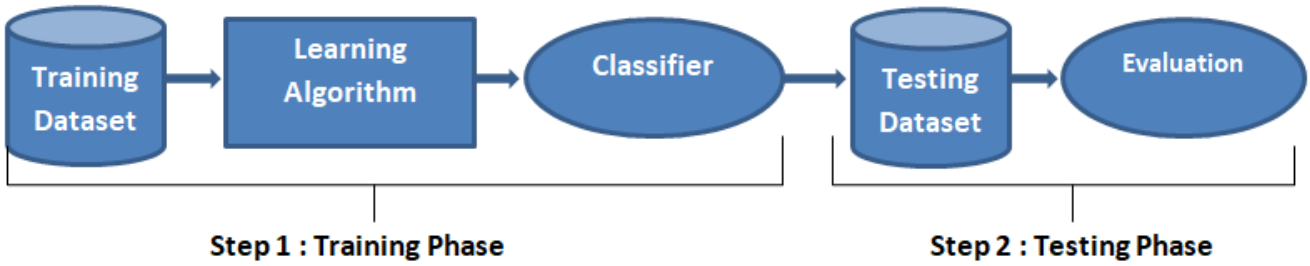

Figure 6: Fundamental Learning Process.

- **Training dataset :** It is a subset of data that is used during the training phase. This data has been labeled with pre-defined classes so that the learning algorithm can learn to produce associations between the data and the corresponding labeled classes.

❑ **Testing dataset :** It is a subset of data that is used during the testing process. it is used to evaluate the classification model generated by the learning algorithm. To maintain the overall veracity of the machine learning algorithm,this data must remain unseen by the algorithm during training.

Algorithms for machine learning are broadly classified into three types: supervised learning, unsupervised learning, and reinforcement learning. In this section, we will briefly review supervised learning algorithms, as they are an important part of this thesis, and then go on to study various foundational algorithms that will help the reader build a solid foundation for understanding more complex algorithms in the field of deep learning.

3.2.2 Supervised learning algorithms

Supervised learning is a type of predictive learning algorithm in which we predict the label of unknown objects using label-based associations inferred by the algorithm during its training phase[Murphy , 2012]. As illustrated in Figure(7), supervised learning is divided into two major branches Regression and Classification, each with its own set of algorithms.

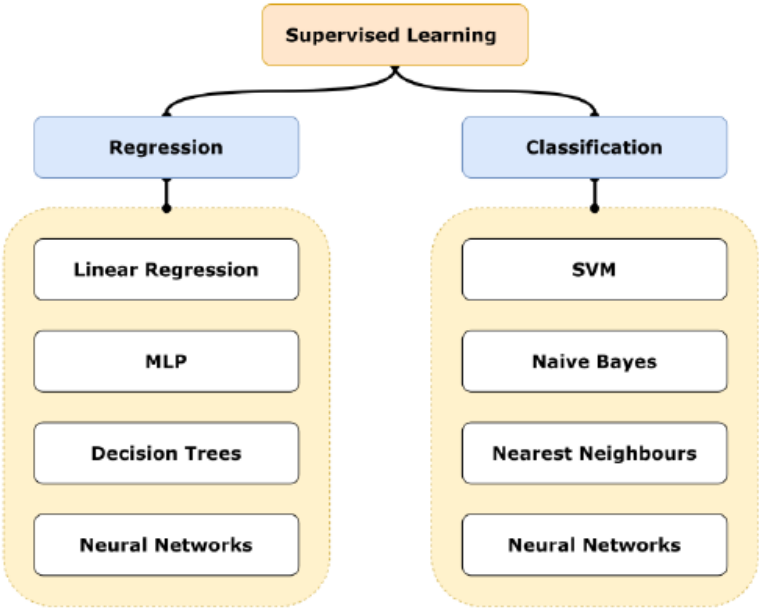

Figure 7: Supervised Learning Taxonomy.

The goal of the algorithm in the supervised learning approach is to learn mappings from input x to output y , given a labeled set of input-output pairs $D = (x_i, y_i)_{i=1}^{N}$, and produce a prediction function. D denotes the training set, and N denotes the number of training examples.
The nature of the training input is determined by the type of problem that the algorithm is attempting to solve. The x_i is a D-dimensional vector or set of numbers that represents the simple features or attributes. x_i, on the other hand, can represent complex structured objects such as an image, time series, e-mail, graphs, etc [Jurafsky , 2019]. Depending on the problem, the output y_i can also take various forms.
If the value of y_i is a categorical variable from a finite set, $y_i \in \{1, ..., C\}$, such as normal or malicious, then the problem is known as classification or pattern recognition. Similarly,when y_i is a real value, the problem is considered as a regression. In simple terms, regression involves predicting a real value, leading to a label estimation whereas, classification involves identifying class membership of a given sample. The function learned during the training phase is also known as a classification model or simply a classifier.

3.2.3 Neural Network

Neural networks are a type of machine learning model that is inspired by neurons in a brain system. In a machine learning context, a neuron is a computational unit with scalar inputs and outputs. Each neuron is also associated with a weight parameter. The neuron multiplies each input unit by its weight, sums all input units, and then applies a nonlinear function to the result to generate an output [Grosan & Abraham, 2011].

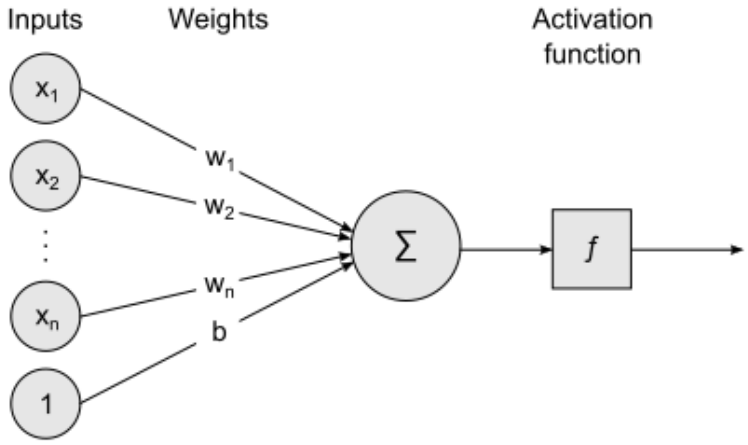

Figure 8: Perceptron Architecture [Hanukoglu,2019]

A perceptron is the simplest neural network architecture, consisting of only two layers of input and output layers, as shown in Figure(8), where we have X_n input units, each with a weight association. We send the input units to the next layer, which, as previously discussed, sums them and uses an activation function, such as a sigmoid function in the case of logistic regression, to find the $\hat{y}$ output based on a boundary decision criterion.

Perceptron classification is limited to linear classification, which means that we can only classify linearly separable sets of vectors. If the vectors are not linearly separable, the perceptron will fail to predict correctly. In contrast, by adding more layers to the perceptron architecture, also known as hidden layers, we progress towards a multi-layered perceptron (MLP) architecture that can perform non-linear classifications and solve much more complex problems.

3.2.4 Multi-layered perceptron

A multi-layered perceptron is a perceptron augmented with additional intermediate layers known as the hidden layers . MLP is a feed-forward neural network, which means that the computation process moves iteratively to the next layers without getting stuck in a loop. The output of each layer becomes the input of the subsequent layers, and no outputs are ever passed back to the previous layers.
The data appears to be moving forward in this manner, so we classify MLP as a feedforward network. The central units of the feedforward MLP are input layer units, hidden layer units, and output layer units. Each layer's units are linked to all the units in previous layers. As a result, the architecture is also known as a fully connected network, with one hidden layer between the input and output layers. Each unit in the hidden layer is connected to x_n units in the input layer, each with a weight association and bias. The hidden layer can be represented as a vector h, the output of which can be written as : Equation (2),

$$h = \sigma(Wx + b). \tag{2}$$

The function represented by σ is an activation function, W is a single matrix of weight associations between the input and hidden layer units, and b is the bias vector for the entire layer. The combination of the weight vector W_{ij} ,which represents the weight of the connectivity between the ith input layer unit and the jth hidden layer unit, into a single matrix W , simplifies and accelerates the computation for the hidden layer in the feedforward network. Furthermore, the output of the hidden layer becomes the input of the output layer. The weight matrix

between these two layers is denoted by the letter U. The output z can now be calculated as follows : Equation (3)

$$z = U \times h. \tag{3}$$

To predict the class labels, we would also need to normalize the output z, which is a vector of real number values, into a probability distribution encoding. In neural networks, we typically use the Softmax function to normalize the output layer, according to the Equation (4) :

$$softmax(z_i) = \frac{\exp(z_i)}{\Sigma_j \exp(z_j)} \tag{4}$$

A softmax function converts logits, which are simply the numerical output of the last linear layer of a multi-class neural network, into probabilities by taking the exponents of each output and normalizing it by the sum of all the exponents. As a result, the vector as a whole adds up to one, providing us with a probability distribution to map the correct prediction labels to.

Neural networks can be thought of as a series of stacked logistic regression classifier units that learn the data representations and induce them into the neural network's subsequent layers. This makes the neural network classifier more powerful in learning data representations on its own, without the need for anyone to handpick the network's feature templates. The self-organization of neural networks distinguishes them from other classical machine learning algorithms. We described a neural network architecture with one hidden layer in this section. Shallow neural networks are the name given to such neural networks. In the following sections, we will discuss architectures that employ multiple hidden layers, also known as deep neural networks.

3.3 Deep learning architectures

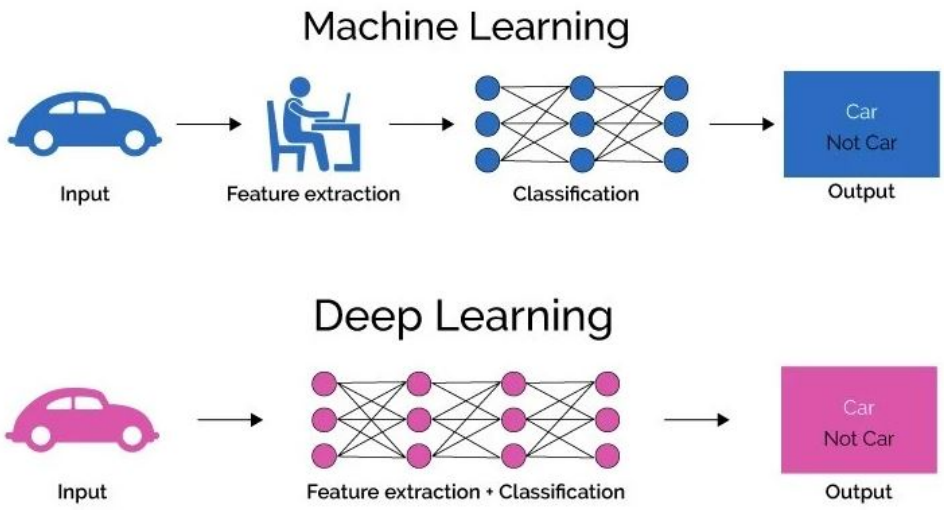

Figure 9: Comparison between machine learning and deep learning classification [Alex,2017]

Deep learning allows computational models made up of many processing layers to learn numerous abstraction levels of data representations .A deep learning model is one that has a large number of completely connected hidden layers, as opposed to shallow learning models, which have only a few hidden layers. The information flow can be used to classify deep neural networks. A feedforward-DNN is a network in which information goes from an input layer to an output layer without any feedback replies. A recurrent neural network, on the other hand, is a neural network architecture that has been integrated to function with several feedback loops.[LeCun, 2015]
The ability of a DNN to learn representations from a raw dataset is one of its most important features. A representation or feature learning is a neural network model's ability to autonomously discover the representations in data required for feature detection and categorization . As illustrated in Figure(9), deep learning networks are used to replace human hand-picking of domain-specific features, which is a requirement for various data mining

and machine learning techniques.[Bengio, 2013]
Deep learning can be simply defined as a class of machine learning algorithms that employ multiple layers of functional units to learn and extract features from a raw dataset. As we progress through the layers, the features extracted become more pronounced, allowing the learning model to infer accurate solutions for the given prediction or classification task. In the following sections, we'll discuss two types of deep neural networks: Convolutional Neural Networks and Recurrent Neural Networks.

3.3.1 Convolutional Neural Network (CNN)

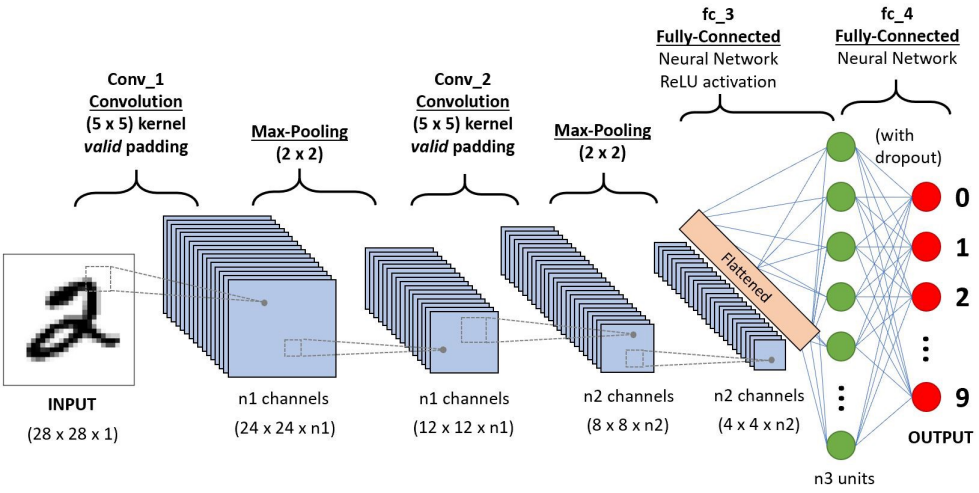

Figure 10: CNN Architecture [Yash,2020]

CNN stands for convolutional neural network, which is a type of feed-forward deep learning network used to solve visual and text-based problems. [Hubel and Wiesel, 1995] pioneered the study of neurons in the visual cortex of mammals to learn how neurons in visual pathways take information from patterns cast on the retina of the eye and modify it on the way to the cerebral cortex, which analyzes and recognizes an image. This discovery influenced the design of [Kunihiko Fukushima, 1988] Neocognitron , a multilevel artificial neural network with cascading layers made up of two components : the S-cell layer and the C-cell layer. In Neocognitron, S-cell layers are the primary feature extraction units, whereas C-cell layers pool information from previous simple cells and give it back to subsequent simple cell layers in a feed-forward manner. With the exception of backpropagation as the principal learning method, a modern CNN is a sequential iteration of Neocognitron architecture. [Yann LeCun et al, 1998]. used the MNIST dataset to demonstrate one of the first implementations of a CNN architecture known as LeNet-5 for the job of hand-written digit recognition. A CNN architecture, as shown in Figure(10), is made up of a stack of different types of layers organized into two main components: a convolution and pooling layers unit that extracts the input layer's features and a fully-connected layer that classifies the results of the preceding feature extraction units into a predictive label output. CNN has a distinct operation for each layer, which is briefly summarized as follows:

a **- Convolution Layer**

The convolutional layer is the central component of a CNN, and it generates feature activation maps from the input layer by moving various receptive fields, also known as filters, across the width and height of the input layer and computing the dot product between the input layer and entries in the filter.
The convolution procedure generates multiple two-dimensional activation maps, as shown in Figure(11), which are then fed into successive pooling layers. The stride value, which is set to one by default, determines how far the filter moves per step. The activation function ReLU is also used in the convolution layer, which turns all negative values to zero.

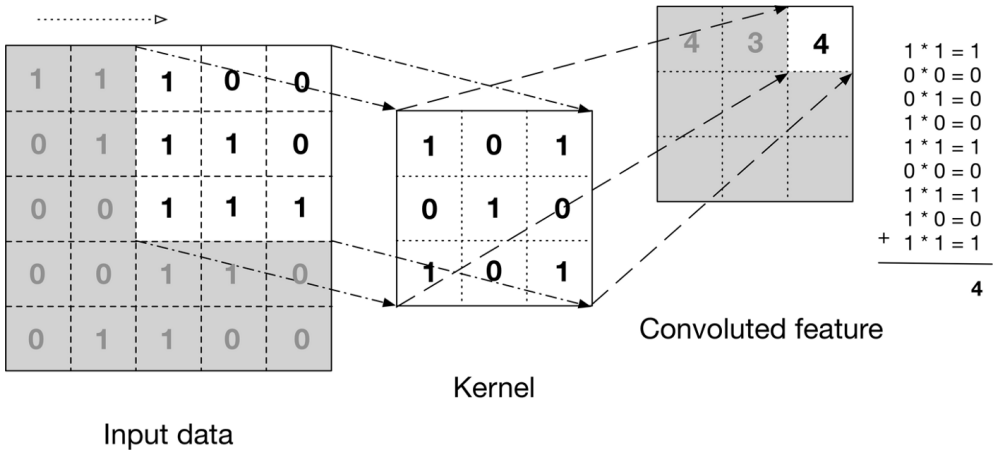

Figure 11: Feature Map computation by Convolution Layer [Li Yin,2018]

b **- Pooling Layer**

The activation volume generated by the convolutional layer is down sampled using the CNN's Pooling Layer. This is critical for decreasing the number of computations. When compared to average or L2-norm pooling, max pooling is the most often utilized method in CNN.
It takes the maximum value of the image in that window, sliding spatially across the full image to get its output. It uses a window of a modest size, 2 or 3, and takes the maximum value of the image in that window. Maximum pooling, average pooling, and sum pooling are all examples of pooling criterion. Figure(12) shows how to use max pooling to subsample the feature map.

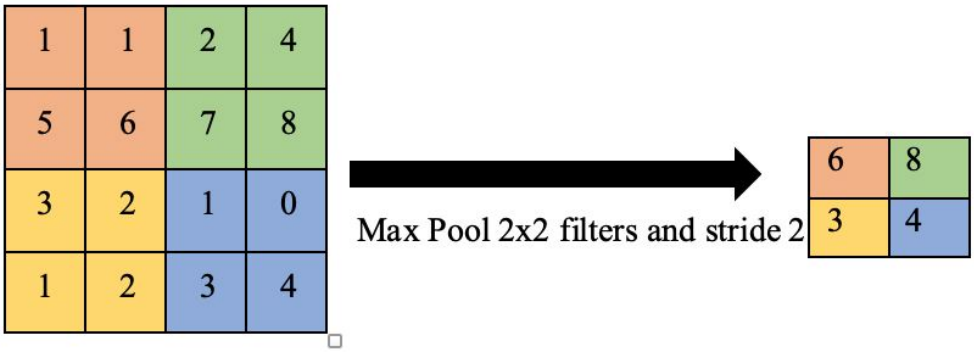

Figure 12: Max Pooling operation with 2x2 Filter and Stride value 2 [Rachmad,2020]

c **- Fully Connected Layer**

The final unit in the CNN architecture is a fully connected layer, similar to the artificial neural networks that were previously considered. All activations in the preceding layer are related to the neurons in this layer. The output of the final pooling layer is flattened into a 1-D vector space after the features are inferred from the input layer's matrix space, as shown in Figure(13). The fully connected layer uses the flattened column vector as input to interpret the features further, which is done by training the network via backpropagation over a number of epochs. The final unit of the fully connected layer generates class label predictions using an activation function such as a sigmoid or softmax activation function, which is also the CNN's final output.

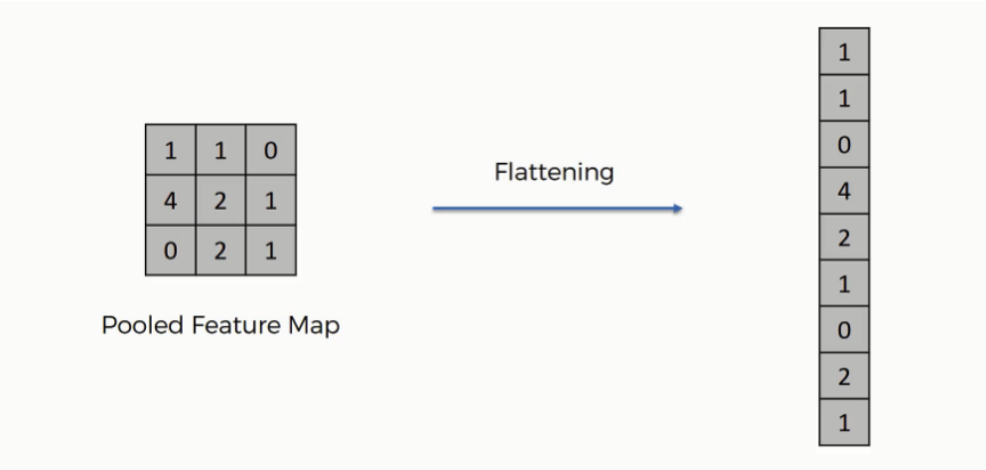

Figure 13: Flatten operation converting feature matrix into 1-D vector input [Rabia,2021]

3.3.2 Recurrent Neural Networks (RNN)

A recurrent neural network (RNN) is a form of artificial neural network that is designed to work with time series or sequence data. Ordinary feed forward neural networks are only designed to handle data items that are unrelated to one another. However, if we have data in a sequence where one data point is dependent on the preceding data point, we must change the neural network to account for these dependencies. RNNs feature a concept of'memory,' which allows them to store the states or information of prior inputs in order to construct the sequence's next output.

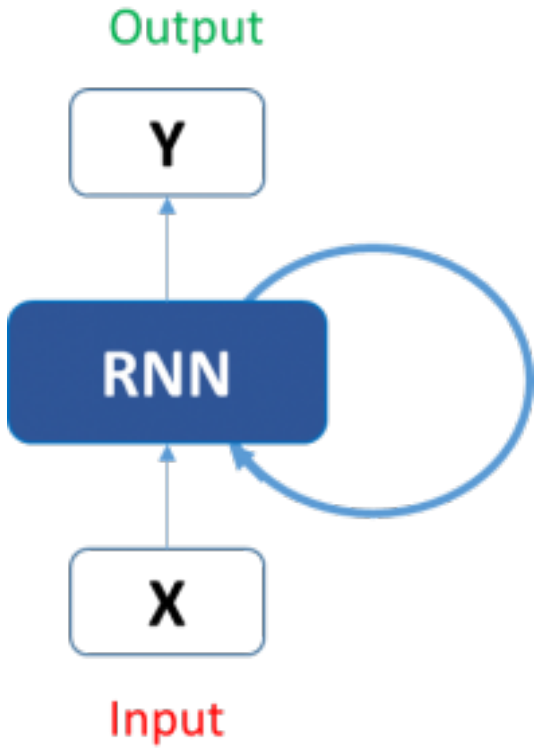

Figure 14: simple RNN cell Architecture [Pranj,2017]

As previously stated, information flows in a single forward direction in typical neural networks. In any sequence

of events, the network does not keep track of its past states. Recurrent Neural Networks (RNNs) are a sort of deep learning architecture that, in addition to feedforward connections, have looping feedback connections that allow the model to store permanent information across time.[Graves, 2012]
An RNN, as shown in Figure(14), receives input x_t at a time stamp t to create y_t, which is the network's output. Moreover, the network computes an internal state at time stamp t, represented by h_t, which it passes from one time step to the next internally within the network, where:

$$h_t = f_w(h_{t-1}, x_t). \tag{5}$$

In this equation, we are computing the recurrence relation in the network at every time step. The value of h_t is determined by function F which is parametrized by a weight w the older state of the network donated as h_{t-1} and the input vector x_t time t.

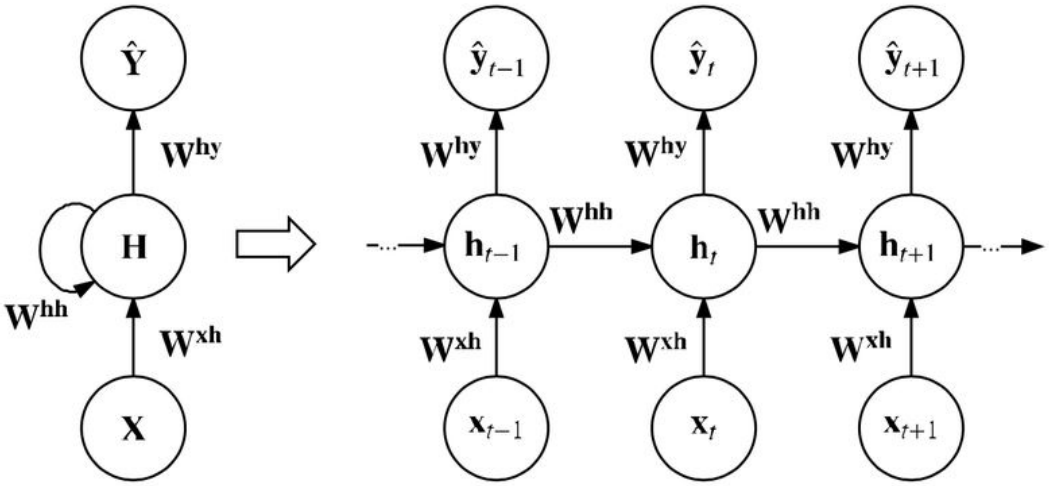

Figure 15: Unrolled RNN Architecture [Khodadadi,2019]

We unroll an RNN to grasp how it computes the output of its network, as shown in Figure(15), where we can explicitly explain the flow of weight matrices that remain the same through the network for a given time step. We also compute the loss value from each RNN unit, capping of a single loop of forwarding across the network. After that, all of the computed loss values from the separate time steps are added together to form a single loss value L, which also determines the network's total loss. Now the updated hidden state of each step in the forward pass can be expressed on equation (6) as follows,

$$h_t = \tanh(W_{hy}^t h_{t-1} + W_{hx}^t x_t) \tag{6}$$

In contrast to a sigmoid function, which only outputs non-negative values, tanh symbolizes the hyperbolic non-linear function employed with RNN, whose value can be both negative and positive, allowing for a drop or increase in states. We employ two weight matrices, W_{hy}^T and W_{hx}^T, as shown in Figure(15), because we are feeding two independent inputs, one from the previous state and the other from the input x_t.Now the output vector for each timestamp is expressed on equation (7) ,

$$\hat{y}_t = W_{hy}^t h_t \tag{7}$$

Where h_t represents the computed hidden state and w_{hy}^t represents the weight matrix between the hidden state and the output unit.

Training an RNN necessitates updating each weight in the network at each time step, for which we employ the backpropagation through time (BPTT) algorithm, in which errors are propagated backward at each individual

step, then across all time steps to the beginning of the data sequence, as shown in Figure(16).

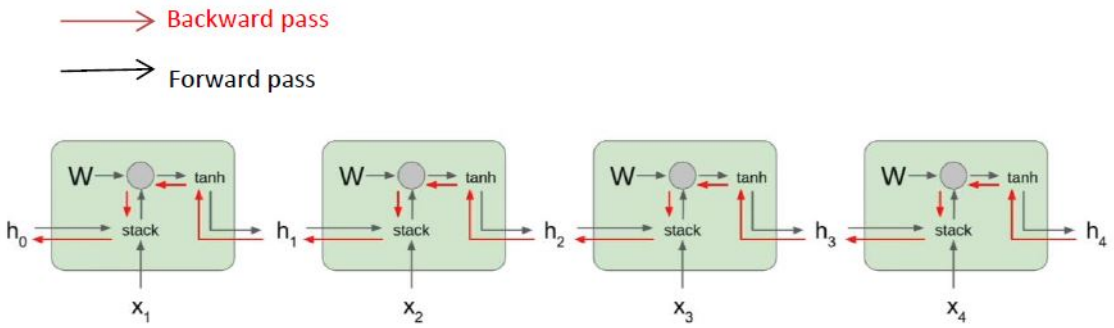

Figure 16: RNN Gradient Flow [Harshit,2019]

Computing the gradient in the network with regard to cell state h_0 in deeper RNN designs requires numerous repeated multiplications of the weight matrix as well as repeated gradient computations using the activation functions. As a result, the network faces the problem of exploding gradients, in which gradients get progressively enormous due to constant accumulation per step and the network is unable to optimize them, resulting in overall network instability due to severe weight changes. Another typical problem with RNN design is vanishing gradients, which occur when gradients grow increasingly smaller after repeated matrix multiplications, preventing the network from being trained and optimized after a certain number of epoch cycles.

RNNs can convert a single input into a single output. In an image classification challenge, such mapping is used. RNNs can also be used to map many inputs into a single output. This mapping could be used to assess the sentiment, whether positive or negative, of a group of words, such as a sentence. Finally, an RNN can convert a series of inputs into a series of outputs. This is commonly used in language translation, where the input is a sequence of text in one language and the output is the translation in another. Figure(17) depicts these mappings.

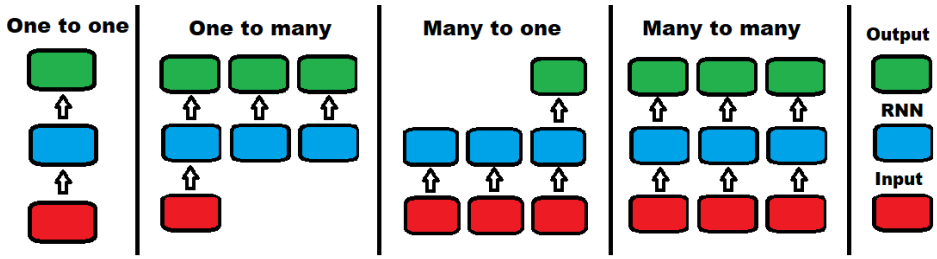

Figure 17: The different input-output mapping used in a recurrent neural network [Rohan,2019]

- **One-to-one**
  With one input and one output, this is the classic feed forward neural network architecture.
- **One-to-many**
  This is referred to as image captioning. We have one fixed-size image as input, and the output can be words or phrases of varying length.
- **Many-to-one**
  This is used to categorize emotions. A succession of words, or even pages of text, is anticipated as input. The result can be a continuous-valued regression output that represents the likelihood of having a favorable attitude.
- **Many-to-many**
  This paradigm is suitable for machine translation, such as that seen on Google Translate. The input could be

a variable-length English sentence, and the output could be a variable-length English sentence in a different language. For video classification at the frame level, the last many to many model might be employed. Expect an immediate output if you feed every frame of a video into the neural network. However, because frames are often dependent on one another, the network must communicate its hidden state from one frame to the next. For this type of task, we'll require a recurrent neural network.

Vanilla Recurrent Neural Networks , Long Short-Term Memory Cells, Gated Recurrent Units, Depth Gated Recurrent Neural Network , and Clockwork Neural Network are examples of recurrent neural networks. For example we list the details of vanilla.

Vanilla RNN

Figure(18) depicts vanilla RNN, which is the most basic kind of RNN. It determines the hidden state using a function f with W parameters. To calculate the current hidden state, the recurrence function, which is most commonly a tanh function, takes the current cell input and the hidden state from a prior time step.

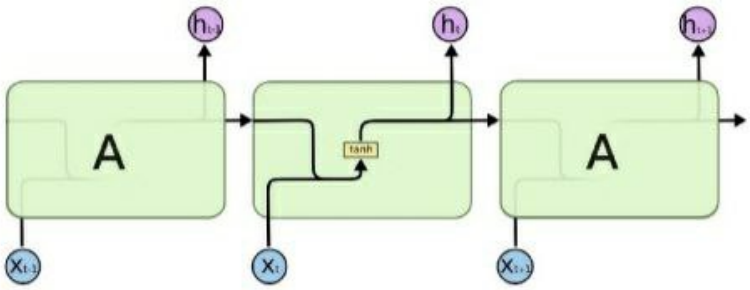

Figure 18: The Vanilla Recurrent Neural Network [colah,2015]

The output at that precise time step is calculated using this hidden state. Equations (8) to (10) depict this process.

$$h_t = f_W(h_{t-1}, x_t) \tag{8}$$

$$h_t = \tan h(W_{hy}.h_{t-1} + W_{xh}.x_t) \tag{9}$$

$$y_t = W_{hy}.h_t \tag{10}$$

For short sequences, when the time delay between relevant information and the location where it is needed is tiny, vanilla RNN perform well. Such networks suffer from vanishing/exploding gradients for lengthy dependencies. For sequences of only 10 or 20 characters, gradients can quickly reach zero. Long backpropagation iterations cause this, which inhibits the network from learning how to model the problem effectively.

3.3.3 Long Short Term Memory (LSTM)

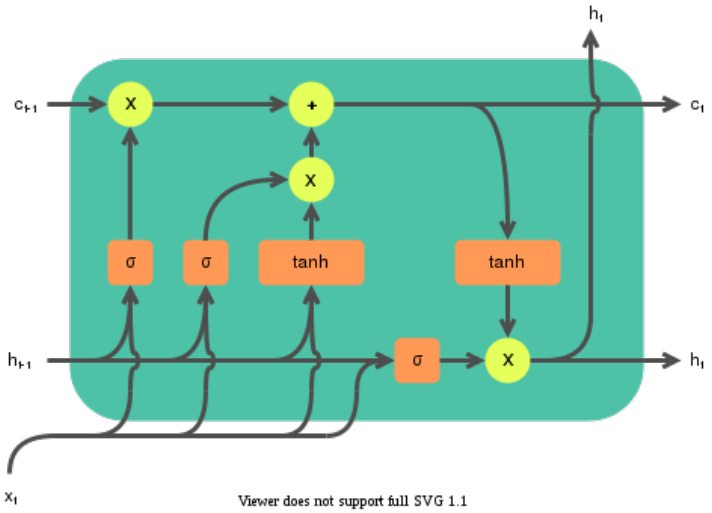

Figure 19: Structure of a LSTM unit [Wikimedia,2020]

[Hochreiter and Schmidhuber, 1997] constructed long short-term memory (LSTM) units that are retrofitted with simple RNN cells to enable them to selectively regulate the information flowing through them in order to alleviate the problem of exploding and vanishing gradients. The information gates, which may selectively add or delete information from their cell state, are the most important component of LSTM units. A sigmoid neural network layer plus a pointwise multiplier unit make up a gate. The sigmoid layer restricts the retention of data passing through the cell between zero and one, thus gating the flow of data. An LSTM unit is made up of three key gate components, as indicated in Figure(19), which are briefly detailed below:

a **- Forget Gate:** This gate specifies which information from the cell state will be discarded. The sigmoid layer makes this judgment, based on the values of h_{t-1} and x_t, and outputs a number between 0 and 1 for the cell state C_{t-1}. The output represents the amount of data that will be stored. A score of 1 indicates that you should keep everything, but a value of 0 indicates that you should completely forget this knowledge.This gate can be expressed by the equation (11)

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{11}$$

b **- Store Gate:**This gate defines the type of data that will be stored in the new cell state. First, a sigmoid layer, also known as the input gate layer, determines which values will be updated in this two-part process. The following layer, tan h, provides a new candidate C_t vector that will be added to the new state.These steps can be expressed by the equations (12) and (13) .

$$I_t = \sigma(W_f.[h_{t-1}, x_t] + b_i) \tag{12}$$

$$C_t = \tan h(W_c.[h_{t-1}, x_t] + b_c) \tag{13}$$

Based on the computation of the last two gates, we now update the old cell state C_{t-1} to the new cell state C_t. We increase the old state f_t, thereby forgetting the earlier information, and then add it to the information from the storage gate, i.e. the value produced from $i_t * C_t$.This step is expressed by the equation (14).

$$C_t = f_t * C_{t-1} + i_t * C_t \tag{14}$$

c **- Output Gate:**Finally, in the current cell state, the cell must determine what information it will output. We decide how much information about the cell state will be outputted using the sigmoid layer of the gate. The cell state is also passed via a tan h unit, which squashes values between -1 and 1, and then multiplied by the sigmoid gate's output.The process can be expressed in the following equations (15) and (16),

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \tag{15}$$

$$h_t = o_t * \tan h(C_t) \tag{16}$$

The major feature of LSTM is its capacity to preserve independence for each cell in the network, which eliminates the problems of vanishing and exploding gradients encountered in simple recurrent neural networks. This allows the network to generate long- and short-term retention dependencies without losing key data or filtering out irrelevant data.

3.4 Conclusion

Machine learning or deep learning has been in great demand in recent decades, and this can be said as a result of the availability of computing power, as well as the availability of data in an organized and large manner. This has made the world today turn towards research and development in this field to benefit as much as possible from the available information to produce new knowledge and expectations that may be crucial for many fields. In this chapter, we discuss many aspects of machine learning, such as deep learning, some types of artificial neural networks, and their improvements to solve problems that researchers face in accessing highly efficient systems. This chapter has been helpful in understanding the different techniques used in the field of machine learning and deep learning, helping us to identify appropriate techniques to implement our vision-based system for detecting falls in the elderly.

Chapter 4

Methodology and System Design

4 Chapter 4 : Methodology and System Design

4.1 Introduction

This chapter describes implementation details of the system. The system can be divided into four modules or steps, an input step that collects data, followed by a pre-procssing step where motion history images are generated, followed by a convolutional neural network for feature extraction, and finally a neural network of densely interconnected neurons that is used for classiffcation. The implementation of the system is done using pytorch [Sarah Lewis,2019] the open source machine learning (ML) framework based on the Python programming language and the Torch library.

The algorithms used by deep neural networks allow learning the parameters of the functions that best classify the data. This eliminates the need for hand-crafted features in order to categorize data, but offers a new difficulty for machine learning programmers. That is, we need to discover the appropriate data format that would allow the network to learn those properties. Therefore, in this chapter, we offer more details on pre-processing and representation of data and description of Dataset, that collectively gave insight into the appropriate data representation for our purpose. Figure (20) shows a diagram of the stages of the fall detection system.

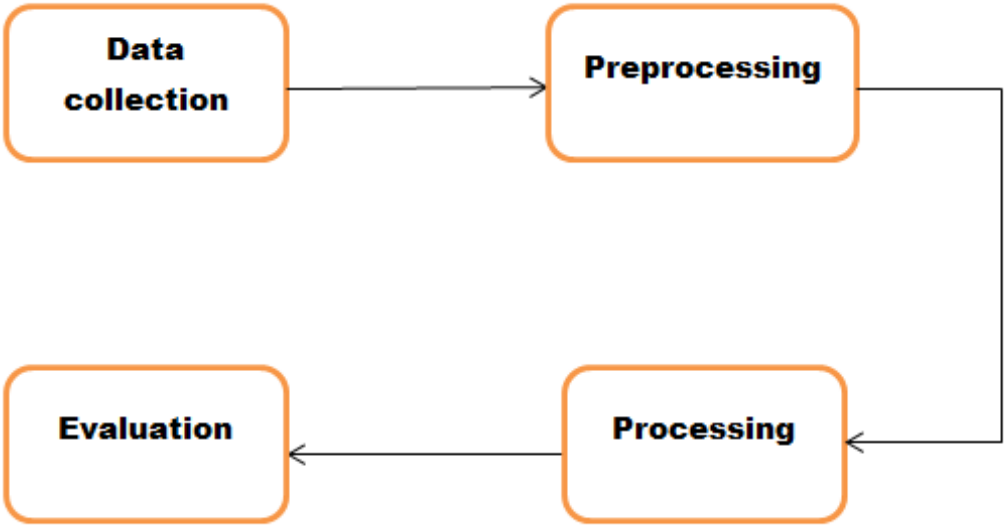

Figure 20: the stages of the fall detection system.

4.2 Data Pre-processing and data representation

4.2.1 Motion tracking

In computer vision, motion tracking is the technique of separating a moving item from the backdrop in a series of photographs. Optical flow and background subtraction, in combination with motion history images, are two such approaches.

a **- Background subtraction**

Background subtraction is a frequent pre-processing step in motion analysis. A moving item can be differentiated from the backdrop using a fixed camera by computing the frame difference between two consecutive frames. The difference function D can be defined as en equation (17).

$$D(x, y, t) =\mid I(x, y, t) - I(x, y, t-1) \mid, \tag{17}$$

where $I(x, y, t)$ is the pixel value at (x, y) at time t for an image I.

b **- Motion history image**

To capture the movement which is considered to be temporal, one technique in computer vision we can use is called Motion History Image (MHI). The concept is to incorporate the previous movement information into the current image by decaying the previous position of the person along a specific period or a number of frames.[Ahad, M et al, 2012]
A motion history image (MHI) is a way to represent motion over time in a single image. It keeps a history of temporal changes for each pixel, which then decays over time. The MHI $H_T(x, y, t)$ can be computed using an update function $\psi(x, y, t)$:

$$\begin{cases} T & if \quad \psi(x,y,t) = 1 \\ max(0, H_T(x,y,t-1) - \delta & otherwise. \end{cases} \tag{18}$$

Here (x, y) and t shows the position and time, $\psi(x, y, t)$ gives the motion in the current image, the duration T decides the temporal extent of the movement (e.g. in terms of frames), and δ is the decay parameter. this function is called for every new frame to be analyzed in the video sequence. the resulting image will be in grayscale where more recently moving pixels are brighter and vice-versa.
The update function ψ can be defined through a function as in equation (19),

$$\psi(x,y,t) = \begin{cases} 1 & if \quad D(x,y,t) > \xi \\ 0 & otherwise. \end{cases} \tag{19}$$

where ξ is a threshold value, that is, the motion is only detected when the difference is larger than ξ. the time complexity of H_T is linearly dependent on the number of pixels in the image. That is, calculating the difference between two images using the background subtraction method defined in equation (19) and then updating each pixel in the MHI, the total cost is : O(N), where N is the number of pixels in the images.

4.2.2 Pre-Processing

The pre-processing step has three important tasks, sampling images from an input stream, re-scaling the sampled images, and finally generating motion history images. these steps are executed in that order, see figure (21).

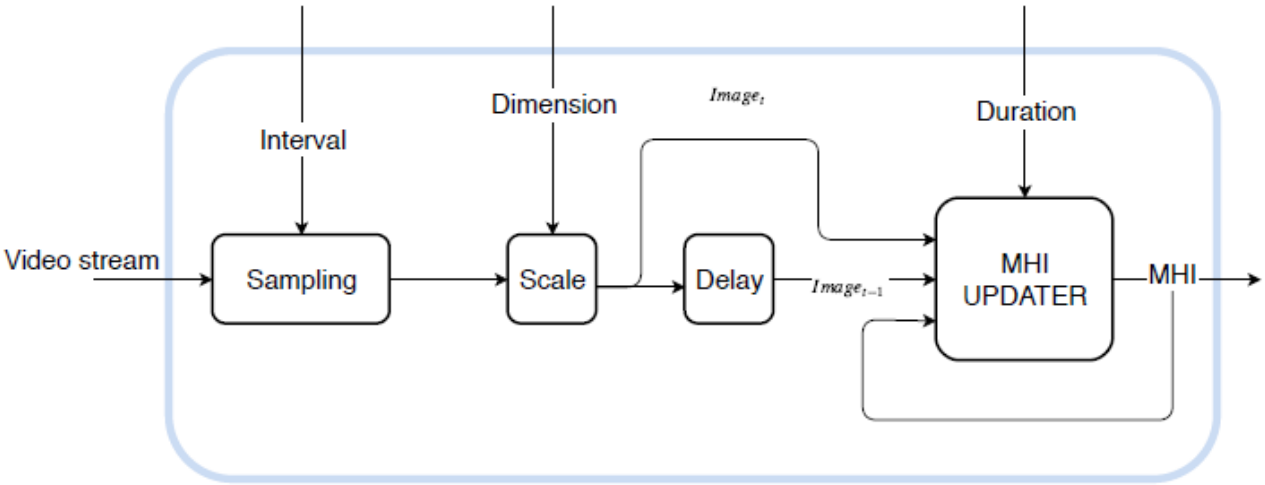

Figure 21: Shows an overview of the pre-processing module. It takes an image stream as input and outputs an motion history image [Haraldsson,2018]

The preprocessing module is in charge of updating the state, which is the MHI "motion history image", whenever a new image is provided to the module. The state-updating-behavior procedure's is determined by three parameters:

- Interval indicates how often images are sampled from the image stream; for example, if interval is two, every second image will be utilized to update the MHI.
- Dimension defines the MHI's width and height in pixels.
- Duration defines the number of frames embedded in the MHI; for example, if duration equals five, the MHI contains the last five sampled images. It's worth noting that the module just has to save the previous image and the current MHI because the prior images can be discarded whenever the MHI is updated.

The module can be adjusted to capture a time period T seconds of video footage in the MHI given an input stream with a given frames-per-second (FPS) using the aforementioned parameters. This is described on equation (20).

$$T = \frac{duration * interval}{FPS} \tag{20}$$

The two variables interval and duration can be used to fine tune the time span, as shown in equation (20). The preprocessing module uses duration = 40, interval = 2, and dimensions = 128 * 128.

Updating MHI:

Figure (22) shows an implementation of the MHI updating function, which is based on H_T, which was defined in equation (20).

Figure 22: Shows generated MHIs. In A) a person is falling from a chair, B) a person is falling from standing position, C) a person walking around.

Below is the pseudocode of the implemented algorithm:

Input : A MHI mhi, the current image I_c, the previous image I_p, a duration δ, and a threshold ξ.
Output: An updated MHI mhi^*.
1 $I_d \leftarrow$ DiffRGB(I_c, I_p);
2 $I_g \leftarrow$ ConvertToGrayScale(I_d);
3 $I_t \leftarrow$ BinaryThreshold(I_g, ξ);
4 $mhi* \leftarrow$ Decay(mhi, δ);
5 $mhi* \leftarrow$ Add($mhi*, I_t$);

The algorithm starts by subtracting the background from two successive sampled images as described in section 4.2.1. The end product might be viewed as a snapshot of the motion between the two images. The algorithm then converts the difference-image I_d to grayscale, with brighter pixels indicating more movement and darker pixels indicating less movement. The pixel values are saved in the range $[0, 1]$, therefore each pixel is compared to a threshold value $\xi = 0.1$ when determining if I_t has moved enough. As a result of this operation, you now have an

image with just black or white pixels. Finally, the MHI's status is updated in two steps.First by reducing all pixel values with $\frac{1}{duration}$, and secondly by adding the binary image I_t to the MHI.

4.2.3 Data Splitting

The captured images from RGB videos were converted from the default bitmap image file to PNG format for easier compatibility with the available image processing libraries. This enabled the easier processing of the images. Further, each of the images was annotated with a fall classification to represent a fallen person. The images were split into images for training and images used for testing the developed model. This was achieved after selecting the images according to the number of people appearing in the image. This ensured the proper distribution of the images in both the training and validation set to reduce the chance of bias. As highlighted earlier, the training set were placed in a folder with their data annotation information. Similarly, a validation set folder was created with the pictures and the annotation data.

4.3 Dataset

We used the Fall Detection Dataset (FDD) [Charfi, 2013] was utilized for training.For training efficiency, not having to pre-process the dataset all the time, the complete FDD was converted to a new dataset called Motion Dataset (MD) that consisted solely of motion history images.

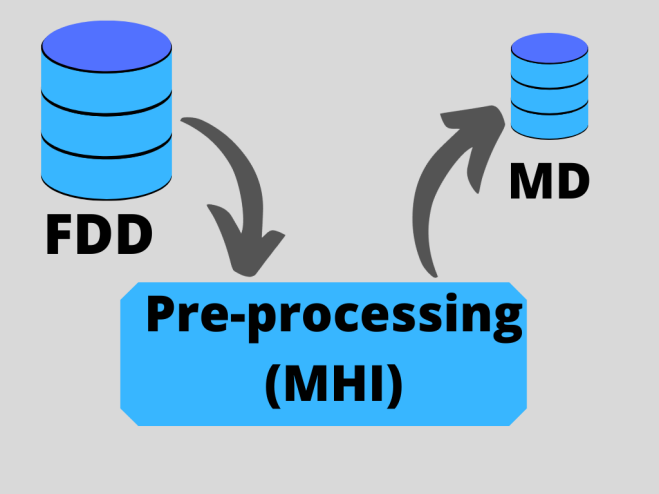

Figure 23: Pre-processing operation.

4.3.1 What is FDD ?

FDD It was recorded using a camera with 25 FPS and a resolution of 320 × 240 pixels ,and comprises of 190 films in different simulated scenarios including offce, coffee room, home, and lecture room. By simulate, it indicates that the situations were staged and folks in the videos were falling on purpose. The falls were done for different angles, falling backwards, forwards or to the sides, and also from different positions such as sitting and standing. Some movies would not involve any fall but rather have similar gestures such as picking up an object from the foor, or sitting down in a sofa or on a chair.

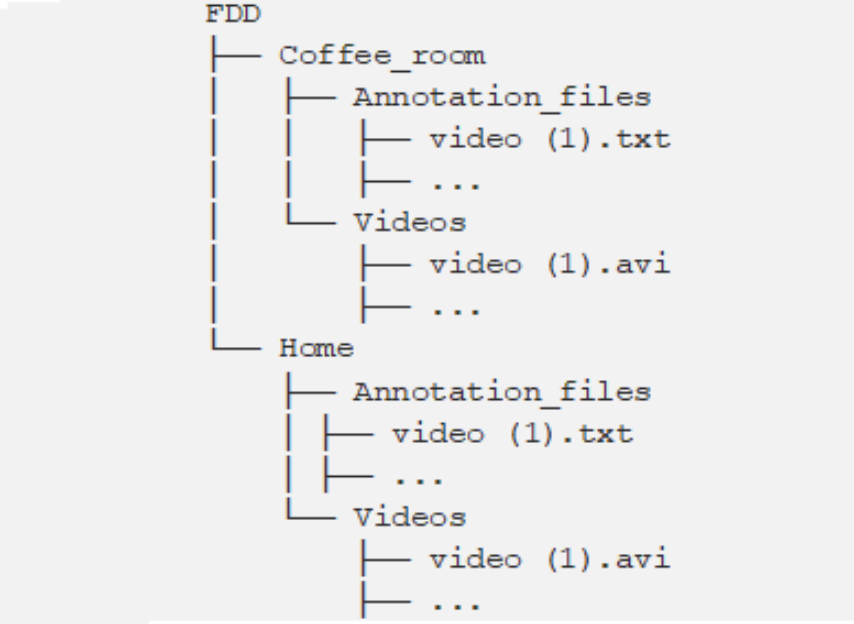

Figure 24: FDD structure.

4.3.2 What is MD ?

MD created by passing the FDD, frame by frame through the preprocessing module and recording the output. The produced MHI was separated into two classes, **fall** and **non-fall** . For MHI belonging to the Fall-class, it must have the most recent photo taken in the critical phase period of the Fall, otherwise the MHI will belong to the non-Fall-class

Figure 25: MD structure.

MD was further separated into three subsets : train, val, and test with a ratio of 0,7, 0,2, and 0,1 respectively. MHIs originating from the same video would be in the same subset to avoid any correlation between the subgroups. The train set was utilized for training, the val set for validation during training and for determining suitable model parameters, and the test set for evaluation.

4.3.3 Dataset description

Table No. (1) shows MD the total number of tires and the number of tires falling and not falling in (FDD ,MD ,train ,val ,test)

Dataset	Size (MB)	Total frames	Fall frames	Non-fall frames
FDD	17408	75911	4908 (6.5%)	71003 (93.5%)
MD	25	26346	11446(43.46%)	14891(56.54%)
train	17	21569	11173(51.8%)	10396(48.2%)
val	5.7	3310	153(4.62%)	3157(95.38%)
test	2.4	2133	120(8.23%)	1338(91.77%)

Table 1: Shows the number of frames that are labeled as fall and non-fall.

The MD that was generated As shown in the Table(1). The dataset is unbalanced because the original dataset, the FDD, contains 75911 frames, 4908 of which are labeled as fall and 71003 as non-fall. Because each MHI might span several frames and hence contain frames labeled as non-fall, there are more MHIs labeled as fall than there are frames labeled as fall in the original dataset.

4.4 Framework's overall architecture

4.4.1 Transfer learning

Transfer learning consists in reassign the parameters of a neural network trained with one dataset and task to another problem with a different dataset and task.When the target dataset is significantly smaller than the base dataset, transfer learning can be a powerful tool to enable training a large target network without over-fitting. We have used MobileNet-v2 as the base model, pre-trained for object detection task on the ImageNet dataset. Learning millions of parameters during this pre-training will be done in order to get generic visual features. These features will be fed to our convolutional neural network .We retrained the network on the FDD dataset [imvia , 2020] to learn more motion features that could be later used to classify falls in the motion stream. Finally, we reuse the network weights and fine-tuning the classification layers in order to generate two classes 'fall' or 'not fall'.

4.4.2 Proposed Neural Network Architecture

To overcome these challenges with CNNs, we offer a more elegant deep learning architecture in this study that can function with less training photos and produce more precise classifications. The fundamental idea is to add layers to a traditional contractual network, replacing pooling operators with up sampling operators. As a result, these layers improve the output resolution [O.Ronneberger, 2015], [S.Guan, 2020].
High resolution characteristics from the contracting path are merged with the up sampled output to localize. Based on this knowledge, a subsequent convolution layer can learn to create a more exact result. One significant change in our architecture is that the up-sampling portion now has a large number of feature channels, allowing the network to pass context information to higher resolution layers.

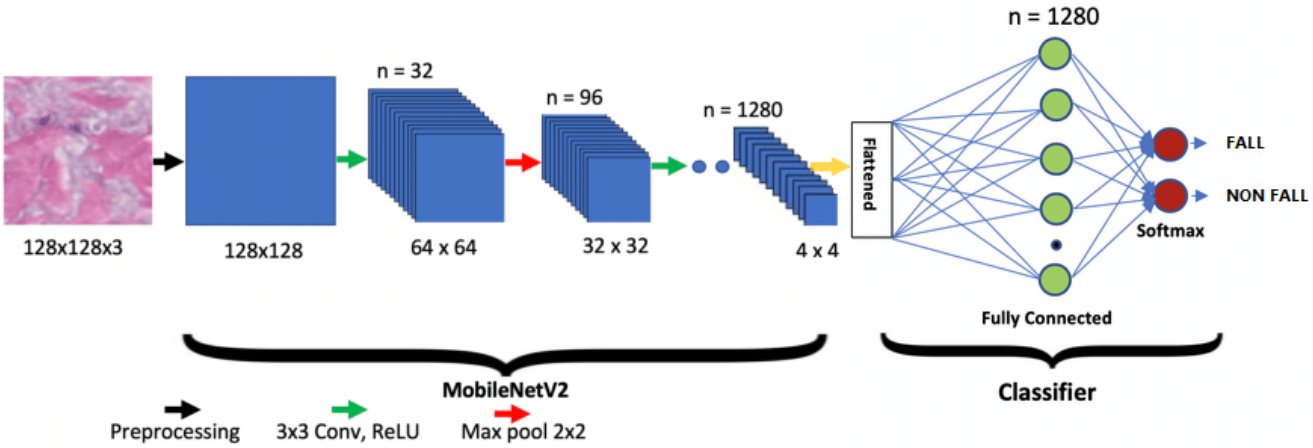

Figure 26: Proposed Neural Network Architecture to classify human fall.

- **MobileNetV2 Stage**

  The proposed network architecture, shown in the figure (26) for binary classification , consists of a contracting path (left side) and a classifier head (right side). Now we describe architecture of The contracting path (MobileNet-v2) in detail. The basic building block is a bottleneck depth-separable convolution with residuals. The detailed structure of this block is shown in Table (2).

Input	Operator	Output
h × w × k	1 × 1 conv2d , ReLU6	h × w × (tk)
h × w × tk	3 × 3 dwise s=s, ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	linear 1 × 1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k'$

Table 2: Bottleneck residual block transforming from k to k0 channels, with stride s, and expansion factor t, and hight h, and width w.

MobileNetV2's architecture (see table (2)) includes a fully convolutional layer with 32 filters, followed by 19 residual bottleneck layers. Because of its robustness when utilized with low-precision computation, we use ReLU6 as the non-linearity [Andrew G et al, 2017]. During training, we always use kernel size 3×3, which is common for current networks, and we use dropout and batch normalization. [Mark Sandler et al, 2018].

We use a consistent expansion rate across the network, with the exception of the initial layer. In our tests, we discovered that expansion rates of 5 to 10 produce almost equal performance curves, with smaller networks performing somewhat better with lower expansion rates and bigger networks performing slightly better with higher expansion rates.[Mark Sandler et al, 2018].
We use an expansion factor of 6 to the size of the input tensor in all of our key studies. The intermediate expansion layer is 64×6 = 384 channels for a bottleneck layer that receives a 64-channel input tensor and generates a tensor with 128 channels.

- **Trade-off hyper parameters**

  We tune our architecture to multiple performance points, as described in [Andrew G et al, 2017], by employing the input picture resolution and width multiplier as configurable hyper parameters that may be altered based on desired accuracy/performance trade-offs. Our core network (width multiplier 1, 224×224) uses 3.4 million parameters and has a computational cost of 300 million multiply-adds. For input resolutions ranging from 96 to 224 and width multipliers ranging from 0.35 to 1.4, we investigate performance trade-offs. The network computational cost ranges from 7 to 585 million multiply additions, with model sizes ranging from 1.7 million to 6.9 million parameters [Mark Sandler et al, 2018].as show on figure (27).

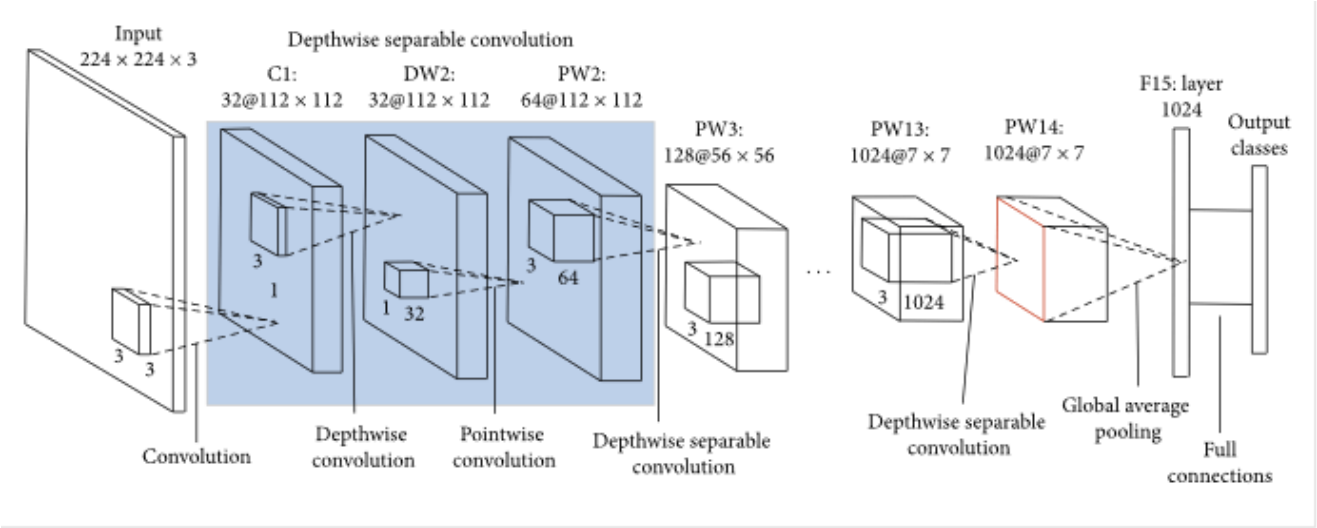

Figure 27: MobileNetV2 convolutional neural network architecture [Wang,2020]

One minor technical difference from [Andrew G et al, 2017] is that we apply the width multiplier to all layers except the very last convolutional layer for multipliers less than one. For smaller models, this increases performance.

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1×1	-	1280	1	1
$7^2 \times 1280$	avgpool 7×7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1×1	-	k	-	-

Table 3: MobileNetV2 : describtion of al sequence.

As shown in the Table (3) Each line describes an n-times repeating sequence of one or more identical (modulo stride) layers. The number of output channels is the same for all layers in the same sequence. Each sequence's initial layer has a stride s, whereas the rest utilize stride 1. kernels 3×3 are used in all spatial convolutions. As shown in Table (2), the expansion factor t is always applied to the input size. [Mark Sandler et al, 2018].

- **Classifier (Fully Connected Layers) Stage**
  At the end of the last block of the MobileNetV2 stage, the feature map matrix is flattened into vector form and fed into a fully connected layer, like a neural network, called the classifier stage. With the fully connected layers, we combined these features together to create a model. Finally, we have an activation function, Softmax, to classify the outputs as a fall case or non fall case.

4.4.3 Training with a Pre-Trained Model

A good initialization of the weights is highly crucial in deep networks with several convolutional layers and diverse pathways through the network. Otherwise, some portions of the network may activate excessively while others

never contribute. The initial weights should ideally be adjusted so that each feature map in the network has approximate unit variance.
Since we work with a medium dataset in fall theme, it is a common practice to take advantage of features learned by a model trained on a larger dataset in the same domain. This is done by instantiating the pre-trained model and adding a fully connected classifier on top. A pre-trained model is a saved network that was previously trained on a large dataset, typically on a large-scale image-classification task. The pre-trained model is "frozen" and only the weights of the classifier are updated during training. In this case, the convolutional base extracted all the features associated with each image, and a classifier was trained to determine the image class given that set of extracted features.
In this study, we used all the model features extracted from theMobileNetV2 model pretrained on the ImageNet dataset with 1.4M images and 1000 classes for all the training parameters in the MobileNetV2 stage of our network architecture [M.Sandler, 2018] , [A. Michele, 2019].

4.4.4 Fine-Tuning a Pre-Trained Model

Along with the training of the new classifier, one technique to improve performance even further is to train (or "fine-tune") the weights of a few selected layers of the pre-trained model. The weights will be forced to be modified from general feature maps to features particular to the dataset during the training phase. The farther up a layer is in a convolutional network, the more specialized it is. The first few layers learn very basic and generic aspects that apply to practically all image formats. The features get more particular to the dataset on which the model was trained as you move higher up. Instead of overwriting the generic learning, finetuning aims to adapt these specialized features to operate with the new dataset.

4.5 Conclusion

In this section ,we pre-processed and description the data ,we also proposed a model based on detection the fall of the elderly using the Python programming language and the Torch library on the fall detection dataset ,and training this model.

Chapter 5

Results and Discussions

5 Chapter 5 : Results and Discussions

5.1 Evaluation of results

In this section we evaluate the system based on the problem definition from the section (1.4). An evaluation of the performance of the system and the accuracy of detecting positive and negative cases of falls is conducted for the total test dataset. We will also compare the quality of the system with other similar systems, based on sensitivity, specificity and accuracy. This will give an insight into the possibility of applying the system to reality based on the results obtained.

5.1.1 Throughput test

Since we did not have a computer with powerful features that would enable us to implement our project, we used the Google Colab service.
Google Colab is a research tool for machine learning education and research , available in two versions free and paid. It is a Jupyter notebook environment that does not require any setup to use. Google Colab works with all major browsers such as Google Chrome, Mozilla Firefox, Safari. Google Colab allows us to use Jupyter notebook Portable and share it with others without the need to download, install or run anything on your computer other than the browser, we used the free version period, but it was very intermittent and had weak features such as GPU and RAM account resources, also little execution time. It wasn't enough to get our work done, so we signed up for the paid Collab Pro service.
We also faced a problem with storage space, as we relied on Google Drive to store files and data collection. So we made a paid subscription to get extra storage space.

Configuration material

GPU: 1xTesla P100, having 3584 CUDA cores, compute 5.3, 16GB HBM2 Vram
CPU: 2 core hyper threaded i.e., (2 core, 4 threads) Xeon Processors @2.20GHz 56MB **Cache RAM**: 13GB available.
high-RAM: 27.3 gigabytes of available RAM.

5.1.2 Detection test

The system must determine if an MHI reflects a fall or not from the perspective of binary classification. The accuracy, sensitivity, and specificity measures are used to evaluate the system's performance. Where sensitivity indicates how well the system detects genuine falls, and specificity indicates how well the system accurately classifies non-falls. Accuracy is simply the ratio of correctly classified to incorrectly classified data. Sensitivity and specificity are particularly valuable since they are not influenced by unbalanced datasets, as are the fall detection datasets. These metrics are defined as:

$$Sensitivity = \frac{TP}{TP + FN} \tag{21}$$

$$Specificity = \frac{TN}{TN + FP} \tag{22}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{23}$$

Where TP = True-Positive, TN = True-Negative, FP = False-Positive, FN = False-Negative. Thus, the system performance is evaluated for each MHI for a given video, that is, we check if a prediction matches the label for a certain MHI. As a result, the following is the definition of the values indicated above:

- TP: a MHI is predicted as fall when labeled as fall.
- TN: a MHI is predicted as non-fall when labeled as non-fall.

- FP: a MHI is predicted as fall when labeled as non-fall.
- FN: a MHI is predicted as non-fall when labeled as fall.

The test was carried out using the dataset MDtest specified in Section 4.3.2, and the result can bee seen in the confusion matrix in table 4. In other words, when a fall occurred over n frames, the algorithm correctly labels 95.61% of these n frames, yet 4.39% of all non-fall frames also get categorized as a fall.

5.1.3 Generality test

The ability of the system to be independent of the background is one of the primary benefits of evaluating motion rather than posture. The system was trained on FDD as outlined in Section 4.4.3 and then tested using the same methods to see if it had learned and generalized the falling motion.

	Actual Fall	Actual No Fall
Pred. Fall	83	27
Pred. No Fall	37	1311

Table 4: Shows the number of TP, FN, TN, and FP for MDtest.

Confusion Matrix

A confusion matrix is a visual representation of the performance of a classification model. It basically is a table with four different combinations of predicted and actual values. A classification model's outcome can be summarized into these four possible categories:

a . True Positive(TP): This corresponds to the values which were predicted to be positive, and they turn out to be positive and correct. In the case of fall detection, the model predicted the person to be fall, and it indeed is fall. Hence the system made a correct prediction. A higher true positive value means the model is making good positive predictions.

b . False Positive(FP): This corresponds to the values which were predictive to be positive, but they turn out to be negative and hence false. In the case of fall detection, the model predicted the person to be fall, but the person a no-fall. A high false positivity of an fall detection leads to unnecessary false alarms and causes needless disruption of services.

c . True Negative(TN): This corresponds to the values which were predicted to be negative and they turn out to be negative and hence correct. In the case of fall detection, the model predicted the person to be no-fall and it was indeed a no-fall. Again, a higher true negative is also deemed to be a positive indicator of the model's performance.

d . False Negative(FN): This corresponds to the values which were predicted to be negative, but they were in actual positive values. In the case of fall detection, the model predicted the person to be no-fall, but the person a fall. This is the most crucial indicator of an intrusion detection system's performance. This value represents how many wrong predictions the model made.

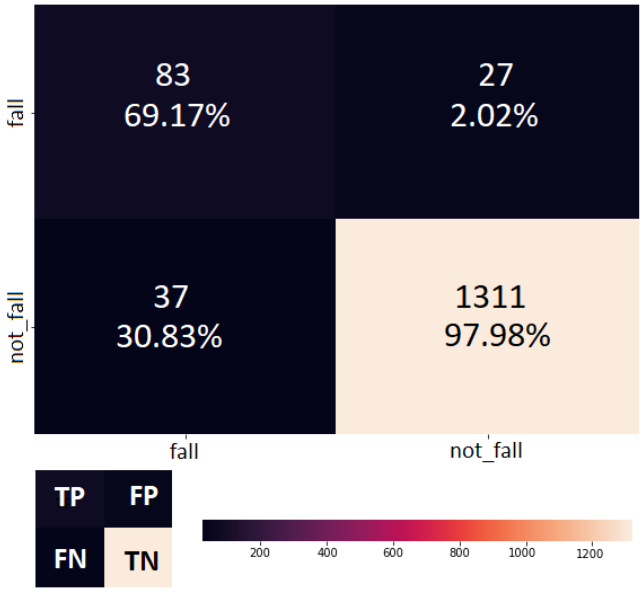

Figure 28: Confusion matrix for fall detection for our model.

In order to evaluate the efficiency of the classification model of our system compared to other fall detection systems that have been trained based on different datasets, we note the results shown in Table (5).

Dataset	Sensitivity	Specificity	Accuracy
URFD	74.90 %	88.20 %	80.20 %
MCFD	33.99 %	88.76 %	83.95 %
MD_{test}	69.17%	97.98%	95.61%

Table 5: Shows the sensitivity, specificity, and accuracy on the three different datasets.

As in the previous section, the detection test, compare on two other public datasets; MCFD [E.Auvinet et al, 2010], and URFD [Bogdan Kwolek, 2014]. The results can be seen in the figure(29).

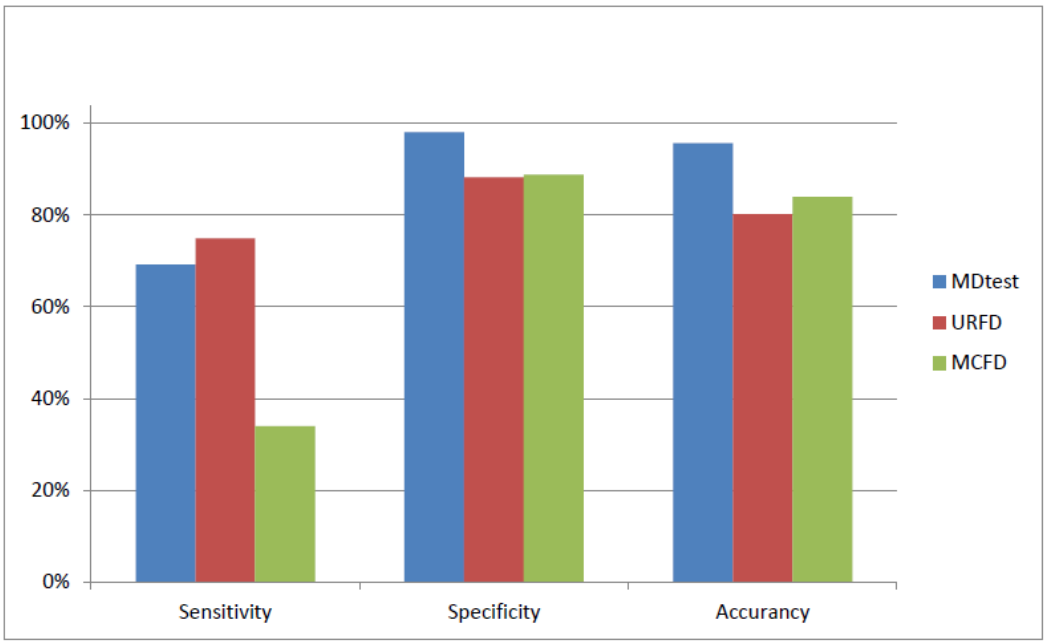

Figure 29: Comparison of the proportions of sensitivity, specificity, and accuracy on the three different datasets.

5.1.4 Fall detection range

When RGB images are used as input to the system, the system can theoretically detect a fall from any distance as long as the resolution is high enough. The FDD used for training simulated a variety of environments, including a standard-sized room with varying distances between the camera and the person falling. There is no information on the dimensions of these rooms, but based on the videos, the areas were around 6×4 meters.

5.1.5 Low intrusiveness

Another requirement was low intrusiveness. Only the silhouette was analyzed, hence this was partially solved using MHI. Furthermore, because the system only analyzes the motion of a person's silhouette, it will not see you while you are still ,such as when you are sleeping.

5.1.6 Interoperability

The output of the neural network is in the interval [0,1], which can be understood as a confidence value for how convinced the system is that it has detected a fall. This might then be used to communicate with other systems, such as an alarm system, to make decisions based on the system's predictions.
Where is efficient automated data sharing between applications, databases, and other computer systems is a crucial component throughout networked computerized systems, especially interoperability in healthcare information and management systems.

5.2 Discussion

In this section, we will compare the performance of our system to that of other similar vision-based systems, as well as evaluate its flaws and potential improvements. We'll also review some problems that came up during the process of building and training the model, before concluding with some comments about deep learning.

Approach	Sensitivity	Specificity	Accuracy
Nunez-Marcos [Nunez-Marcos et al, 2017]	99.00 %	97.00 %	97.00 %
Charif [Charfi, 2013]	98.00 %	99.69 %	99.61 %
Zerrouki and Houacine [N, Zerrouki and A, Houacine, 2018]	-	-	97.02 %
Our Proposal	69.17%	97.98%	95.61%

Table 6: Comparison between our system and other approaches that used FDD for evaluation.

5.2.1 Comparison

When comparing the results of Section 5.1.3 with those of other vision-based systems, one issue is the absence of public available datasets, which has pushed many to construct their own. As a result, our system will be compared against other vision-based systems that have utilised the FDD.Table (6) shows the results from Section 5.1.3 compared to others on the FDD.

- Nunez-Marcos [Nunez-Marcos et al, 2017] produced state-of-the-art classification results. They evaluated their system in the same way that we did, frame by frame classification. They employed the more complicated VGG network as a classifier, and the amount of trainable parameters for VGG and MobileNet with α= 1 may be compared in [Andrew G et al, 2017], with VGG having 138 million compared to 0.5 million for MobileNet. We used $\alpha = 0.25$ and hence have a lower level of complexity. Their research made no mention of whether their system method works in a real-time context, however they did train on an NVIDIA TitanX.

- Charif [Charfi, 2013] used a majority vote over a series of frames to test their system. They classified 18 images individually before making a final decision in their case. Their findings are striking, demonstrating that a fall detection system based on simpler classification methods is more effective than deep neural networks.

- Zerrouki and Houacine tested their method on sub-video sequences that were labeled as fall or non-fall. However, they merely reported the accuracy, which might be misleading if the dataset is unbalanced, as in the case of FDD [N, Zerrouki and A, Houacine, 2018].

5.2.2 System drawbacks & Improvements

Even though the system has shown to be fairly effective, there is still potential for development.

a **- Input**

The use of RGB photos as input is an obvious flaw in the current design. Because the preprocessing module cannot identify a moving person from the background if the input feed is all black, the system cannot function effectively at night. Instead of using daylight-independent cameras, infrared cameras or other comparable daylight-independent cameras could be used.

b **- Pre-processing**

How the MHIs are generated is a problem linked with brightness. the background subtract function set in is sensitive to changes in light, for example, if a light is turned on, all pixels will move and the resulting MHI will be covered with just white pixels. Other, more complicated backdrop subtraction algorithms, such as removing shadows, partially overcome the difficulties with light changes. Because light changes in the input stream are an inherent part of RGB cameras, one could argue that the problem is at the input level, which could be rectified for example by employing a time-of-flight camera.

c **- Feature extractor & Classifier**

During training, it was often found that the network would overfit on the training data. Which is called overfitting. To solve this problem, the network must distinguish silhouette poses, not just the person's silhouette. And a larger dataset might be another way, to minimize overfitting.
The present classifier has only been trained on FDD, and each video in that dataset only has one actor. The system is incapable of detecting many falls at the same time. but could be tackled via image segmentation, then analyze it.
Decision are made is currently done at a frame-by-frame, which could result in isolated decisions. Charif

[Charfi, 2013] shown that utilizing a majority vote approach to solve this problem yielded good results.
The feature extractor's input layer currently accepts a 3-channeled input, but the resulting MHI are only 1-channeled. The grayscale images were copied over all three channels to solve this problem. the depth of the input layer should be reduced to one.

d **- Problems encountered**

During the early stage of development, after selected FDD as the dataset, it was observed that the annotation of FDD was incomplete. Less than half of all the videos were annotated, and we had no access to the tools that had been used for labeling. Because it is desired to have an uniform labeling method for the entire dataset.
One aspect that have affected the whole project is the time demanding task of training neural networks, in particular the convolutional layers. It could take days merely to test one set of parameters.
There were several reasons for using a pre-built network, such as MobileNet, rather than building one from scratch. Using a network that has been shown to converge well towards an optimal solution seemed like a reasonable choice, and it was used in [Nez-Marcos et al, 2017] where they used VGG. It was also discovered early on that using weights learned from ImageNet to transfer learning gave a significant boost in classification accuracy, but training a custom network from scratch on ImageNet would be too time consuming for this project's time window. As a result, using a model with pre-trained weights was appropriate because it saved time.
The limited amount of public fall detection datasets has already been noted as a barrier to deep learning solutions in the field of automatic fall detection [Shehroz,S et al, 2017]. Fortunately, transferred learning can help to alleviate this issue to some extent, although a larger and more diverse dataset with additional regions and people would be ideal. Other strategies include data augmentation, which involves transforming data, such as rotating and scaling, to produce a huge dataset. This was tried, however it resulted in faster overfitting. The explanation for this could be that the transformations were insufficient; for example, because turning a photo 90 degrees would cause the person to tumble off the wall, we utilized considerably less rotation, resulting in nearly similar images. Perhaps simply arranging the photos horizontally, so that a person descending from the left would fall from the right, and vice versa, might suffice. This strategy was abandoned because data augmentation effectively increases data size and therefore training time.

e **- Our thoughts**

We believe this study was a fascinating study that demonstrates how strong, but also constrained, deep learning can be. By examining silhouettes of people, the network became surprisingly good at recognizing falls. As a result, deep learning is a truly revolutionary technology that will likely have a profound impact on our entire society. With the rise of the internet of things, it will be interesting to observe if machine learning algorithms can be implemented at the network's edge in the next years. That is, rather than being done in data centers, intelligent decision making is done at the sensor level.

5.3 Conclusion

In this work, we presented a real-time vision-based fall detection solution. The system captures temporal features using motion history images, that are used as input to a deep convolutional neural network. The network employs depthwise separable convolutional layers to effciently extract spatial features from the motion history images. The system performed fairly well against previously proposed systems in terms of sensitivity and specificity on the public fall detection dataset FDD. The system was also evaluated on two other public datasets to test the generality of the system. However, it was shown that the system is very dependent on the training data and could not generalize the moving motion of a fall across different datasets.

Chapter 6

Conclusion

6 Chapter 6 : Conclusion

6.1 Conclusion

Because late detection of falls can have serious consequences such as death or serious injury, especially for the elderly and sick. Fall detection systems seek to reduce response time as much as possible.
In this work, we present a solution for fall detection, which is a vision-based system. The system uses the public fall detection dataset (FDD) to train a binary classification model to classify the activity as fall or no fall. The system first processes the input data, by means of a preprocessing unit, which converts RGB videos into motion history images MHI, which are passed as input to a deep convolutional neural network CNN. The network ingeniously extracts the spatio-temporal features of the falling motion from motion history images MHI.
It was noted that the performance of the system is somewhat good in terms of sensitivity and specificity compared to other previously proposed systems that depend on the fall detection dataset FDD. The system showed good accuracy in detecting falls but it relies heavily on training data and cannot generalize the movement of falls to different datasets. The preprocessing unit can be set manually to adjust the sampling frequency and capture longer or shorter time intervals, both parameters can be adopted to improve the accuracy of the system. The pre-processing algorithm has some flaws such as light and background effect, but nevertheless it was a partial solution to the problem of interfering with people's privacy, where silhouette images are processed instead of raw input data. The use of transfer learning ,by pre-trained model MobileNetV2 has been very successful , especially when a large amount of training data is not available.

6.2 Delimitations

Although various fall detection systems exist, our project will focus on creating a vision-based system. The major objective will be to develop a machine learning model that can identify falls in pictures, such as video broadcasts. As a result:

- We won't be able to set up a complete system with cameras and computation units.
- The system part that alerts a third party if someone falls will not be built.
- To train the model, we'll use a pre-processed dataset (images), which means we won't finish pre-processing the video clips.

6.3 Future Vision

In the future, researchers should look into transferable learning from other domains, such as video classifications. There are still a few things to finish, such as To use a real camera, as the current input module is simulating a camera using stored videos, an Application Programming Interface (API) to integrate with other systems, and further development to make the choice more robust, i.e. higher sensitivity and specificity, Which can be accomplished by following the recommendations:

- To improve the accuracy of posture detection, more balancing data for falls and routine activities should be used to train the model.
- Data augmentation settings can be explored further, such as rotating the person to a larger angle and using vertical and horizontal flips to improve the model's learning for detection. In the final step, better posture detection can enhance LSTM classification accuracy.
- It's also a good idea to training the model in low-light situations at night. This may aid in the model's subsequent generalization.
- To explore more possibilities in fall detection, compare the detected region features with various classification algorithms.
- In the future, it may be possible to track a falling person while another person is present.

References

[Aziz et al, 2011] : Aziz, O.; Robinovitch, S.N. An Analysis of the Accuracy of Wearable Sensors for Classifying the Causes of Falls in Humans. IEEE Trans. Neural Syst. Rehabil. Eng. 2011, 19, 670–676.

[Alwan et al, 2006] : Alwan, M., Rajendran, P. J., Kell, S., Mack, D., Dalal, S., Wolfe, M., & Felder, R. (2006). A smart and passive floor-vibration based fall detector for elderly. In Information and Communication Technologies (Vol. 1, pp. 1003-1007).

[Agrawal et al, 2017] : Agrawal, S.C.; Tripathi, R.K.; Jalal, A.S. Human-fall detection from an indoor video surveillance. In Proceedings of the 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Delhi, India, 3–5 July 2017; pp. 1–5.

[Auvinet et al, 2011] : Auvinet, E.; Multon, F.; Saint-Arnaud, A.; Rousseau, J.; Meunier, J. Fall detection with multiple cameras: An occlusion-resistant method based on 3-D silhouette vertical distribution. IEEE Trans. Inf. Technol. Biomed. 2011, 15, 290–300.

[Angal et al, 2016] : Angal, Y.; Jagtap, A. Fall detection system for older adults. In Proceedings of the 2016 IEEE International Conference on Advances in Electronics, Communication and Computer Technology (ICAECCT), Pune, India, 2–3 December 2016; pp. 262–266.

[A. Michele, 2019] : A. Michele, V. Colin, and D. D. Santika, "MobileNet convolutional neural networks and support vector machines for palmprint recognition," Procedia Comput. Sci., vol. 157, pp. 110–117, 2019

[Andrew G et al, 2017] : Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR, abs/1704.04861, 2017.

[Ahad, M et al, 2012] : Ahad, M., Rahman, A., Tan, J. K., Kim, H., Ishikawa, S. (2012). Motion history image: its variants and applications. Machine Vision and Applications, 23(2), 255-281.

[Belshaw et al, 2011] : Belshaw, M., Taati, B., Giesbrecht, D., & Mihailidis, A. Intelligent vision-based fall detection system: preliminary results from a real world deployment. RESNA/ICTA, 5-8.

[Bagalà et al, 2012] : Bagalà, F.; Becker, C.; Cappello, A.; Chiari, L.; Aminian, K.; Hausdorff, J.M.; Zijlstra,W.; Klenk, J. Evaluation of Accelerometer- Based Fall Detection Algorithms on Real-World Falls. PLoS ONE 2012, 7, e37062.

[Brulin et al, 2012] : Brulin, D.; Benezeth, Y.; Courtial, E. Posture recognition based on fuzzy logic for home monitoring of the elderly. IEEE Trans. Inf. Technol. Biomed. 2012, 16, 974–982.

[Bian et al, 2015] : Bian, Z.P.; Hou, J.; Chau, L.P.; Magnenat-Thalmann, N. Fall detection based on body part tracking using a depth camera. IEEE J. Biomed. Health Inform. 2015, 19, 430–439.

[Bengio, 2013] : Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence, 35(8), 1798-1828.

[Bogdan Kwolek, 2014] : Bogdan Kwolek and Michal Kepski. "Human fall detection on embedded platform using depth maps and wireless accelerometer". In: Computer Methods and Programs in Biomedicine 117.3 (2014), pp. (489-501). [Last Accessed 24 April 2022] Url: http://www.sciencedirect.com/science/article/pii/S0169260714003447.

[Charfi, 2013] : Imen Chari et al. / Optimized spatio-temporal descriptors for real-time fall detection: Comparison of support vector machine and Adaboost-based classification". In: 22 (Oct. 2013), pp. 041106.

[Diraco et al, 2010] : Diraco, G.; Leone, A.; Siciliano, P. An active vision system for fall detection and posture recognition in elderly healthcare. In Proceedings of the 2010 Design, Automation & Test in Europe Conference & Exhibition (DATE 2010), Dresden, Germany, 8–12 March 2010; pp. 1536–1541.

[E.Auvinet et al, 2010] : E. Auvinet et al. cameras fall dataset". In: (2010).

[Feng et al, 2016] : Feng, G.; Mai, J.; Ban, Z.; Guo, X.; Wang, G. Floor Pressure Imaging for Fall Detection with Fiber-Optic Sensors. IEEE Pervasive Comput. 2016, 15, 40–47.

[Girshick et al, 2014] : R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014.

[Girshick et al, 2015] : R. Girshick, J. Donahue, T. Darrell, and J. Malik. Regionbased convolutional networks for accurate object detection and segmentation. TPAMI, 2015.

[Girshick, et al, 2017] : Girshick, R.; Ren, S.; He, K.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Trans. Pattern Anal. Mach. Intell. 2017, 39, 1137–1149.

[Grosan & Abraham , 2011] : Grosan, C., & Abraham, A. (2011). Intelligent systems. Berlin: Springer.

[Graves, 2012] : Graves, A. (2012). Supervised sequence labelling. In Supervised sequence labelling with recurrent neural networks (pp. 5-13). Springer, Berlin, Heidelberg.

[H.Alkittawi, 2017] : ALKITTAWI, Hend. A deep-learning-based fall-detection system to support aging-in-place, p19, (2017). [Last Accessed 03 June 2022]. URL : https://tamucc-ir.tdl.org/handle/1969.6/2975

[Hubel and Wiesel, 1995] : Hubel, D. H. (1995). Eye, brain, and vision. Scientific American Library/Scientific American Books.

[Hochreiter and Schmidhuber, 1997] : Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[imvia , 2020] : imvia Fall Detection Dataset FDD (2020). [Last Accessed 15 April 2022]
Url: https://imvia.u-bourgogne.fr/basededonnees/fall-detection-dataset.html

[Jansen et Deklerck, 2006] : Jansen, B., & Deklerck, R. Context aware inactivity recognition for visual fall detection. Pervasive Health Conference and Workshops, 2006 (pp. 1-4). IEEE.

[Janice Morse, 2008] : Morse, Janice M. Preventing patient falls. Springer Publishing Company, 2008.

[Kido et al, 2009] : Kido, S.; Miyasaka, T.; Tanaka, T.; Shimizu, T.; Saga, T. Fall detection in toilet rooms using thermal imaging sensors. In Proceedings of the 2009 IEEE/SICE International Symposium on System Integration (SII), Tokyo, Japan, 29 January 2009; pp. 83–88.

[Kaur, 2017] : Kaur, R., & Kaur, P. D. (2017). Review on Fall Detection Techniques based on Elder People. International Journal of Advanced Research in Computer Science, 8(3).

[Kunihiko Fukushima, 1988] : Fukushima, K. (1988). Neocognitron: A hierarchical neural network capable of visual pattern recognition. Neural networks, 1(2), 119-130.

[Lídia Sànchez-Riera et al, 2017] : Sànchez-Riera L, Wilson N. Fragility Fractures & Their Impact on Older People. Best Pract Res Clin Rheumatol. 2017 Apr;31(2):169-191.

[Lai et al, 2011] : Lai, C.; Chang, S.; Chao, H.; Huang, Y. Detection of Cognitive Injured Body Region Using Multiple Triaxial Accelerometers for Elderly Falling. IEEE Sens. J. 2011, 11, 763–770.

[Lustrek et al, 2015] : Lustrek, M.; Gjoreski, H.; Vega, N.G.; Kozina, S.; Cvetkovic, B.; Mirchevska, V.; Gams, M. Fall Detection Using Location Sensors and Accelerometers. IEEE Pervasive Comput. 2015, 14, 72–79.

[Li et al, 2012] : Li, Y.; Ho, K.C.; Popescu, M. A microphone array system for automatic fall detection. IEEE Trans. Biomed. Eng. 2012, 59, 1291–1301.

[LeCun, 2015] : LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.

[Lenise A ,2011] : Lenise A. Cummings-Vaughn and Julie K. Gammack. /Falls, Osteoporosis,and Hip Fractures". In: Medical Clinics of North America 95.3 (2011). [Last Accessed 01 June 2022] url: http://www.sciencedirect.com/science/article/pii/S0025712511000174.

[Morse et al, 1987] : Morse, Janice M., Suzanne J. Tylko, and Herbert A. Dixon. "Characteristics of the fall-prone patient." The Gerontologist 27.4 (1987): pp516-522.

[Medicina, 2015] : Self-reported consequences and healthcare costs of falls among elderly women .Alekna V., Stukas R., Tamulaityte-Morozoviene I., Surkiene G., Tamulaitiene M. (2015) Medicina (Lithuania), 51 (1) , pp. 57-62.

[Mirmahboub et al, 2013] : Mirmahboub, B.; Samavi, S.; Karimi, N.; Shirani, S. Automatic monocular system for human fall detection based on variations in silhouette area. IEEE Trans. Biomed. Eng. 2013, 60, 427–436.

[Ma et al, 2014] : Ma, X.; Wang, H.; Xue, B.; Zhou, M.; Ji, B.; Li, Y. Depth-based human fall detection via shape features and improved extreme learning machine. IEEE J. Biomed. Health Inform. 2014, 18, 1915–1922.

[Murphy , 2012] : Murphy, K. P. (2012). Machine learning: a probabilistic perspective. MIT press.

[M.Sandler, 2018] : M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 4510–4520.

[Mark Sandler et al, 2018] : Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520).

[Nizam et al, 2016] : Nizam, Y., Mohd, M. N. H., & Jamil, M. M. A. A study on human fall detection systems: Daily activity classification and sensing techniques. International Journal of Integrated Engineering, 8(1).

[Nunez-Marcos et al, 2017] : Núñez-Marcos, A., Azkune, G., & Arganda-Carreras, I. Vision-based fall detection with convolutional neural networks. Wireless Communications and Mobile Computing, 2017.

[N, Zerrouki and A, Houacine, 2018] : Nabil Zerrouki and Amrane Houacine. "Combined curvelets and hidden Markov models for human fall detection". In: Multimedia Tools and Applications 77.5 (2018), pp. 6405–6424.

[Ozcan et al, 2016] : Ozcan, K.; Velipasalar, S.; Varshney, P.K. Autonomous Fall Detection with Wearable Cameras by Using Relative Entropy Distance Measure. IEEE Trans. Hum.-Mach. Syst. 2016.

[O.Ronneberger, 2015] : O.Ronneberger et al., "U-Net, convolutional neural networks for image segmentation," in MICCAI 2015: Med. Image Comput. Computer-Assisted Intervention – MICCAI 2015, pp. 234–241, 2015

[Pannurat et al, 2017] : Pannurat, N.; Thiemjarus, S.; Nantajeewarawat, E. A Hybrid Temporal Reasoning Framework for Fall Monitoring. IEEE Sens. J. 2017, 17, 1749–1759.

[Pierlenoi et al, 2016] : Pierleoni, P.; Belli, A.; Maurizi, L.; Palma, L.; Pernini, L.; Paniccia, M.; Valenti, S. A Wearable Fall Detector for Elderly People Based on AHRS and Barometric Sensor. IEEE Sens. J. 2016, 16, 6733–6744.

[Poonsri et al, 2018] : Poonsri, A.; Chiracharit, W. Improvement of fall detection using consecutive-frame voting. In Proceedings of the 2018 International Workshop on Advanced Image Technology (IWAIT), Chiang Mai, Thailand, 7–9 January 2018; pp. 1–4.

[Rougier et al, 2006] : Rougier, C., & Meunier, J. Fall detection using 3d head trajectory extracted from a single camera video sequence. In First International Workshop on Video Processing for Security (VP4S-06), June (pp. 7-9).

[Rougier et al, 2007] : Rougier, C., Meunier, J., St-Arnaud, A., & Rousseau, J. Fall detection from human shape and motion history using video surveillance. Advanced Information Networking and Applications Workshops, 2007, AINAW'07. 21st International Conference on (Vol. 2, pp. 875-880). IEEE.

[Ren et al, 2015] : S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards 14 real-time object detection with region proposal networks," in Neural Information Processing Systems (NIPS), 2015.

[Rucco et al, 2018] : Rucco, R.; Sorriso, A.; Liparoti, M.; Ferraioli, G.; Sorrentino, P.; Ambrosanio, M.; Baselice, F. Type and Location of Wearable Sensors for Monitoring Falls during Static and Dynamic Tasks in Healthy Elderly: A Review. Sensors 2018, 18, 1613.

[Rubenstein LZ , 2006] : Rubenstein LZ. Falls in older people: epidemiology, risk factors and strategies for prevention. Age Ageing. 2006 Sep;35 Suppl 2:ii37-ii41. doi: 10.1093/ageing/afl084. PMID: 16926202.

[Sposaro and Tyson's, 2009] : Sposaro, F.; Tyson, G. iFall: An Android application for fall monitoring and response. In Proceedings of the Annual International Conference of the IEEE on Engineering in Medicine and Biology Society (EMBC), Minneapolis, MN, USA, 3–6 September 2009.

[Sixsmith and Johnson, 2004] : Sixsmith, A., & Johnson, N. A smart sensor to detect the falls of the elderly. IEEE Pervasive computing, (2), 42-47.

[Sabatini et al, 2016] : Sabatini, A.M.; Ligorio, G.; Mannini, A.; Genovese, V.; Pinna, L. Prior-to- and Post-Impact Fall Detection Using Inertial and Barometric Altimeter Measurements. IEEE Trans. Neural Syst. Rehabil. Eng. 2016, 24, 774–783.

[Su et al, 2015] : Su, B.Y.; Ho, K.C.; Rantz, M.J.; Skubic, M. Doppler radar fall activity detection using the wavelet transform. IEEE Trans. Biomed. Eng. 2015, 62, 865–875.

[Shiba et al, 2017] : Shiba, K.; Kaburagi, T.; Kurihara, Y. Fall Detection Utilizing Frequency Distribution Trajectory by Microwave Doppler Sensor. IEEE Sens. J. 2017, 17, 7561–7568.

[Sarah Lewis,2019] : PyTorch Meaning and Uses. [Last Accessed 25 April 2022]
Url : https://www.techtarget.com/searchenterpriseai/definition/PyTorch

[S.Guan, 2020] : S. Guan, A. A. Khan, S. Sikdar, and P. V. Chitnis, "Fully dense UNet for 2-D sparse photoacoustic tomography artifact removal," IEEE J. Biomed. Health Inform., vol. 24, no. 2, pp. 568–576, Feb. 2020.

[Shehroz,S et al, 2017] : Shehroz S. Khan and Jesse Hoey. "Review of fall detection techniques: A data availability perspective". In: Medical Engineering & Physics 39 (2017), pp. 12–22.

[Taguchi et al, 2016] : Taguchi, C. K., Teixeira, J. P., Alves, L. V., Oliveira, P. F., & Raposo, O. F. F. (2016). Quality of life and gait in elderly group. International archives of otorhinolaryngology, 20(3), 235-240.

[WHO, 2021] : World Health Organization. Falls. [Last Accessed 29 April 2022]
http://www.who.int/mediacentre/factsheets/fs344/en/

[Wu et al, 2015] : Wu, F., Zhao, H., Zhao, Y., & Zhong, H. Development of a wearable-sensor-based fall detection system. International journal of telemedicine and applications, 2015, 2.

[Wang et al, 2014] : Wang, J.; Zhang, Z.; Li, B.; Lee, S.; Sherratt, R.S. An enhanced fall detection system for elderly person monitoring using consumer home networks. IEEE Trans. Consum. Electron. 2014, 60, 23–29.

[WHO ,2008] : World Health Organization. WHO global report on falls prevention in older age. 2008.

[Yu et al, 2012] : Yu, M.; Rhuma, A.; Naqvi, S.M.;Wang, L.; Chambers, J. A posture recognition based fall detection system for monitoring an elderly person in a smart home environment. IEEE Trans. Inf. Technol. Biomed. 2012, 16, 1274–1286.

[Yann LeCun et al, 1998] : LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

[Habib, 2014] : Smartphone-Based Solutions for Fall Detection and Prevention: Challenges and Open Issues - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Common-basic-architecture-of-fall-detection-and-fall-prevention-systems_fig1_261837566 [last accessed 17 Jun, 2022]

[Nizam, 2016] : Fall detection FPGA-based systems: A survey - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Fall-detection-classification_fig1_311393453 [last accessed 17 Jun, 2022]

[Girshick,2014] : Computer Vision and the Internet of Things Ecosystem in the Connected Home - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/R-CNN-Region-with-Convolutional-Neural-Networks-features-Architecture-Taken-from-9_fig2_330093035 [last accessed 17 Jun, 2022]

[Girshick,2015] : Learning visual models for person detection and action prediction - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/5-Fast-RCNN-object-detection-pipeline-Girshick-2015_fig6_327800511 [last accessed 17 Jun, 2022]

[Hanukoglu,2019] : Hanukoglu, Moshe & Goldberg, Nissan & Rovshitz, Aviv & Azaria, Amos. (2019). Learning to Conceal: A Deep Learning Based Method for Preserving Privacy and Avoiding Prejudice.

[Alex,2017] : Geological Facies Prediction Using Computed Tomography in a Machine Learning and Deep Learning Environment - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Difference-between-traditional-machine-learning-and-deep-learning-Alex-2017_fig8_326834670 [last accessed 17 Jun, 2022]

[Yash,2020] : Deep Learning model-based Multimedia forgery detection - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Architecture-of-CNN-20-The-architecture-of-a-CNN-is-as-follows-Input-layer-The_fig2_346036530 [last accessed 17 Jun, 2022]

[Li Yin,2018] : A Summary of Neural Network Layers. Available from: https://medium.com/machine-learning-for-li/different-convolutional-layers-43dc146f4d0e [last accessed 17 Jun, 2022]

[Rachmad,2020] : SIBI (Sistem Isyarat Bahasa Indonesia) translation using Convolutional Neural Network (CNN) - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Illustration-of-max-pooling-with-filter-size-2x2-and-stride-2_fig4_338845633 [last accessed 17 Jun, 2022]

[Rabia,2021] : Tourist Guide via Image Processing Techniques - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Feature-Map-of-Flattening-Layer_fig4_354866263 [last accessed 17 Jun, 2022]

[Pranj,2017] : Essentials of Deep Learning : Introduction to Long Short Term Memory. Available from: https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/ [last accessed 17 Jun, 2022]

[Khodadadi,2019] : ChOracle: A Unified Statistical Framework for Churn Prediction - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/An-RNN-unrolled-through-the-time-The-same-structure-is-repeated-at-adjacent-time-steps_fig1_335854778 [last accessed 17 Jun, 2022]

[Harshit,2019] : The gradient problem in RNN. Available from: https://kharshit.github.io/blog/2019/01/04/the-gradient-problem-in-rnn [last accessed 17 Jun, 2022]

[Rohan,2019] : 6 Deep Learning models — When should you use them? From ANNs to AutoEncoders. Available from: https://towardsdatascience.com/6-deep-learning-models-10d20afec175 [last accessed 17 Jun, 2022]

[colah,2015] : Understanding LSTM Networks. Available from: https://colah.github.io/posts/2015-08-Understanding-LSTMs/ [last accessed 17 Jun, 2022]

[Wikimedia,2020] : Internal structure of a LSTM (Long Short-term Memory) cell, Opensource LSTM image by Wikimedia. Available from: https://commons.wikimedia.org/wiki/File:LSTM_cell.svg [last accessed 17 Jun, 2022]

[Haraldsson,2018] : Real-time Vision-based Fall Detection : with Motion History Images and Convolutional Neural Networks. Available from: https://www.semanticscholar.org/paper/Real-time-Vision-based-Fall-Detection-%3A-with-Motion Haraldsson/56d51fed57290ff62356694e9311ba035fe131ba/figure/9 [last accessed 17 Jun, 2022]

[Wang,2020] : Wei Wang, Yutao Li, Ting Zou, Xin Wang, Jieyu You, Yanhong Luo, "A Novel Image Classification Approach via Dense-MobileNet Models", Mobile Information Systems, vol. 2020, Article ID 7602384, 8 pages, 2020.