2019

A Vision-based approach to fall detection for elderly patients receiving home-based care

Andrew K. Waruguru
Faculty of Information Technology (FIT)
Strathmore University

A VISION-BASED APPROACH TO FALL DETECTION FOR ELDERLY PATIENTS RECEIVING HOME-BASED CARE

Waruguru, Andrew Kinyua

Master of Science in Information Technology

2019

A VISION-BASED APPROACH TO FALL DETECTION FOR ELDERLY PATIENTS RECEIVING HOME-BASED CARE

Waruguru, Andrew Kinyua

Submitted in partial fulfillment of the requirements for the award of the Degree of Master of Science in Information Technology at Strathmore University.

Faculty of Information Technology

Strathmore University

Nairobi, Kenya

June, 2019

Declaration

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the proposal contains no material previously published or written by another person except where due reference is made in the thesis itself.

ANDREW KINYUA..................................

....

12/7/2019.....................................

Approval

The thesis of Waruguru Andrew Kinyua was reviewed and approved by the following:

Prof. Ismail Ateya D.Sc.,

Associate Professor, Faculty of Information Technology,

Strathmore University

Dr. Joseph Orero,

Dean, Faculty of Information Technology,

Strathmore University

Prof. Ruth Kiraka,

Dean, School of Graduate Studies,

Strathmore University

Abstract

Falls present one of the unintentional accidents for people in the world. The adverse effects of a fall vary with the nature of the fall and the impact with the ground or object. Essentially, falls rarely occur in the daily activities of healthy individuals. The occurrence results in fatal or non-fatal falls. However, the falls are consequential for the elderly people since they result in future related problems or death. As such, elderly patients require additional attention in the case of fall events. Therefore, to mitigate the effect of a fall on an elderly patient, there must be the provision of a fast response mechanism. Response time to medical emergencies plays a key role in patient survival and recovery. As such, medical personnel strive to reduce the response time. Proper and immediate notification of an emergency aids in reducing the response time. In order to substantially reduce the negative effect of the fall or increase the survival chances, patients ought to receive fast medical response. Therefore, the need of a fast and proper notification method that aims at providing relevant information in regards to the nature of emergency of the patient. As such, proper monitoring leads to a reduced response time. Arguably, elderly patients require urgent medical care in case of a fall. This research work proposes a multi-person fall detection system, which implements a vision-based approach for fall detection leveraging on region-based convolution neural network. A fixed camera serves as the input device to capture images of people. The system analyses the image to identify the posture and orientation of the people present in the image. Based on the provided image, the system then classifies the occurrence as a fall or non-fall using the developed model. If it identifies a fall, an alert is then sent to a concerned party. The system achieves a mean average precision of 0.8 in fall detection. Further, the system detects a fall in an image in 3.8 seconds thus improving the response time of the medical personnel to aid in curbing the negative effects of a fall on a patient.

***Keywords**: Falls, response time, vision-based, image-preprocessing, fall detection, Mask R-CNN*

Table of Contents

List of Figures

List of Tables

List of Equations

List of Abbreviations/Acronyms

CDC	–	Center for Disease Control
CNN	–	Convolution Neural Network
GMM	–	Gaussian Mixture Model
GPU	–	Graphics Processing Unit
HMM	–	Hidden Markov Model
ICD	–	International Classification of Diseases
MHI	–	Motion History Image
MMS	–	Multimedia Message Service
RAM	–	Random Access Memory
RCNN	–	Region-Based Convolution Neural Networks
RPN	–	Region Proposal Network
SMS	–	Short Message Service
UML	–	Unified Modelling Language
VCPU	–	Virtual Central Processing Unit
WHO	–	World Health Organization

Acknowledgements

I would like to sincerely thank my supervisor Professor Ismail Ateya for his unending support, patience and dedication in the course of my research study. Notably, his guidance and commitment helped me a great deal in achieving the research objectives. I would like to thank Henry Muchiri for his support in gathering hardware resources required for the study. I would also like to show appreciation to the rest of the Faculty of Information Technology (FIT) members for their comments and recommendations.

Dedication

This research work is dedicated to my beloved parents and my young brother for the support they have provided me through many hardships to achieve academic excellence. I thank God for His abundant grace throughout this academic journey.

Chapter One: Introduction

1.1 Background

Proper health care is vital for the existence of human beings. Internal or external injuries present a major problem for people due to the related cost incurred and the state of being unwell. According to the World Health Organization (WHO), falls are categorized as the second leading cause of accidental or unintentional injury deaths globally (World Health Organization [WHO], 2018). Notably, the organization states that in the Republic of China, for every death that results from a fall, there are twenty-four cases of hospitalization that lasts between 1-9 days, 13 cases that require hospital admission for more than 10 days, 4 cases of permanent disability, and 690 cases of people seeking or lacking medical assistance (WHO, 2018).

Furthermore, the falls occur in low and middle-income countries and result in 60% of the deaths (WHO, 2018). Essentially, people with an age of over 60 years are most likely to die or develop a serious injury as a result of the falls. Notably, this is due to their susceptibility to illness as a result of their decreasing immunity levels. As such, falls affect their daily activities in a significant way if not properly handled by the medical personnel. Arguably, to properly handle the elderly patients' adequate monitoring services need to be provided to the homes.

Patient monitoring promotes the survival chances of recovering patients while it ensures the utilization of different monitoring avenues. Furthermore, a falling patient requires urgent and proactive medical response. Pons et al. (2005) asserts that the response time of less than four minutes significantly affects the survival chances of a patient. Blackwell and Kaufman (2002) argue that the survival chances of a patient only increase with a response time of less than five minutes. Notably, a threshold to the minimum response time of 8 minutes exists such that past the response time, it does not affect the outcome of the patient (Pons et al. 2005; Blackwell & Kaufman, 2002). The senior people in the community have a higher chance to get injuries due to a fall than the young people. As such, the seniors require comprehensive monitoring of their movements. Noteworthy, the number of elderly people falling due to other complicated illnesses has risen significantly (Abreu, Mendes, Monteiro, & Santos, 2012). Therefore, the need to develop solutions that cover the elderly people and other patients.

Currently, different systems enable patient monitoring with the use of biosensors or mechanical sensors. The biosensors monitor the different organs of the body to provide information to the system. As such, the instances are procedural therefore a fall is depicted after its occurrence. However, the cases of falling patients occur randomly therefore not comprehensively covered. For instance, home-based care presents a challenge to families since the patient's must be closely monitored as the falls occur randomly. On the other hand, mechanical sensors provide the capability to detect the occurrence of falls. The systems present a more comprehensive and better concept in detecting and fall notification.

1.2 Problem Statement

The prevalence of falls among the elderly patients present a major health problem to the society. According to the WHO (2016), the global life expectancy has increased by 5.5 years between 2000 and 2016. Moreover, the Center for Disease Control (CDC) asserts that the fall death rates in the United States have increased by 30% from 2007 to 2016 for older adults ("Center for Disease Control and Prevention," 2017). Khow and Visvanathan (2017) note that the falling frequency increases with aging. Arguably, an increase in global life expectancy and an increase in falling frequency result in an increase in falls.

Notably, the falls may result in physical injury and psychological trauma to the patients. Essentially, one out of five falls results in a serious injury such as a head injury ("Center for Disease Control and Prevention," 2017). Therefore, close monitoring would be necessary to increase the medics' responsiveness. Furthermore, home-based care poses a significant challenge to caretakers who must keep a close watch to prevent any adverse outcomes of falls even when they are engaged and out of the sight of their patient. Poor responsiveness can result from these unforeseen falls that can aggravate the health conditions of the seniors (Fleming & Brayne, 2008). Thus, an accurate fall detection mechanism can improve the welfare of the patient with increased responsiveness by the caretakers.

1.3 Aim

The research aims at developing a vision-based system that can detect and identify a fall thereby notifying the medics or concerned parties. The system needs to distinguish between a fall and the daily activities undertaken by the patient. Furthermore, the system must incorporate a notification system that notifies the medics.

1.4 Research Objectives

(i) To review characteristics of a human being falling.

(ii) To investigate the challenges associated with a human being falling.

(iii) To appraise existing models, algorithms and frameworks that can be applied in vision-based fall detection.

(iv) To design a prototype for fall detection using a vision-based approach.

(v) To test the developed prototype.

1.5 Research Questions

(i) What characteristics depict that a human being is falling?

(ii) What challenges arise after a human being has fallen?

(iii) Which existing models, algorithms and frameworks can be applied in vision-based fall detection?

(iv) How can a vision-based approach prototype be developed?

(v) How can the developed prototype be validated?

1.6 Justification

Firstly, the results of the research provide patients with a safety tool or mechanism to ensure that they are closely monitored hence leading to a healthier lifestyle. Notably, acknowledging the number of deaths occurring due to falls the system aims at reducing the number of deaths by providing a higher chance of survival for the patient. As a result, this increases the chances of survival of the patients. Secondly, the research provides a robust method of detecting falls for patients or people thus increasing the efficiency of the medics or healthcare providers. Essentially, this leads to an overall acceptance of the medical fraternity and better healthcare provision for elderly patients. Thirdly, by modelling the necessary framework for fall detection, industrial experts can implement the framework to other sectors of the economy. As such, the framework or prototype can form the basis for fall detection in other fields. Lastly, the study plays a vital role in contributing to research and innovation for the academic professionals. It aims at providing insights currently not in the literature thereby important for future researchers in the field.

1.7 Scope and Limitation

The proposed system detects a fall occurrence and sends a notification. The system processes the information received by the sensor and classify if the activity represents a fall or a normal task (non-fall). The system's environment comprises of a closed surveillance system where the patient can be monitored by a fixed camera. Further, the system only detects the fall due to inactivity of the object in relation to body shape change.

Chapter Two: Literature Review

2.1 Introduction

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 Fall Definition

A fall is commonly visualized rather than defined. The World Health Organization (WHO) outlines a fall as an accidental occurrence that results in a human being coming to rest on the ground, floor or any other lower level. Essentially, in the International Classification of Disease -9(ICD -9) falls are coded as E880-E888 and (ICD-10) as W00-W19. The classification includes falls on the same level and upper levels.

2.3 Falls

Falls present a problem in the health industry. Essentially, falls can be classified as fatal or non-fatal falls. Globally, about 646,000 die as a result of the fall incidents each year depicting the severity of the occurrence (WHO, 2018). Notably, this places falls as the second cause of unintentional deaths in the world. Falls among the elderly people have increased resulting in increased death rates. According to the Center for Disease Control, falls represent the number one cause of fatal and non-fatal injuries in Americans above the age of 65 ("Centers for Disease Control and Prevention," 2016). Additionally, Yoshida asserts that the rate of frequency of falls increases with an increase in age (Yoshida–Intern, 2007). Yoshida (2007) presents that per 1000 population, the injury rate increases from 35 to 76 as the age of the people increases in 65-69 to 80 and above.

Notably, the falls serve as the underlying problem that leads to future complications. Elderly people experience injuries such as broken hips to traumatic brain injury. Mostly, the incidences of shock affect the livelihood of the elderly as they are more prone to the incidence three-fold after the fall. The condition results in hospitalization of the seniors thereby increasing chances of infection by other diseases such as pneumonia. Noteworthy, the CDC states that the trauma of the brain because of the fall serves as the cause of the death in 41 % of the fatalities (Centers for Disease Control and Prevention, 2016). On the other hand, non-fatal falls lead to other complications such as depression or fear of falling. Additionally, the condition worsens with an increase in number of falls of an individual.

Falls occur mostly during the day when people are more active. Essentially, 80% of the falls occur during the day while 20 % occur during the night (Yoshida–Intern, 2007). Notably, the falls in older people occurring at night occur while navigating to the washrooms. Additionally, older people residing in home-based care are more likely to experience a fall as they adapt into a new room or ward (Yoshida–Intern, 2007). Since the falls occur during the day, the major location of the falls occurs outside the home in places such as the yard, street or public place. Essentially, the falls that occur in the home happen mostly in the kitchen, living room and bedroom. Notably, fewer falls occur on the stairs or step stools. The figure below shows the location distribution:

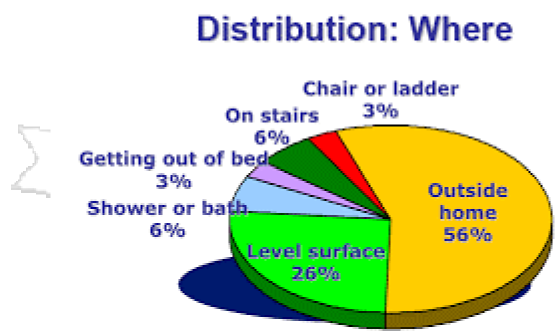

Figure 2.1: Location Distribution (Yoshida-Intern, 2007)

2.3.1 Importance of Fall Detection in the Society

2.3.1.1 Healthcare Costs

According to the CDC, over 3 million patients are treated as a result of a fall related injury. Furthermore, the number of hospitalized patients due to fall related injury reached 800,000 ("Centers for Disease Control and Prevention", 2016). Additionally, the CDC asserts that there over 27000 fatal deaths for senior citizens ("Centers for Disease Control and Prevention", 2016). As such these results to increased costs due to the injuries.

Notably, in 2014 the estimated annual cost of falls accounted to $31 billion ("Centers for Disease Control and Prevention", 2016).

Falls result in physical injury, psychological trauma or death. Terroso et al (2014) assert that elderly patients tend to encounter psychological trauma as a result of a fall. Thus, this affects the lifestyle of the patient.

2.3.1.2 Prevalent Injuries

Non-fatal falls often result in injuries on the patient. As highlighted earlier, elderly patients are more susceptible to falls than younger persons, therefore, more prone to injuries associated with falls. Furthermore, due to ageing, the negative impact of a fall on an elderly patient is higher than in younger population. Essentially, about 95% of the hip fractures reported in hospitals result from falls and eventually lead to the hospitalization of the patients ("Centers for Disease Control and Prevention", 2016).

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 Katsoulis (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 TBI. As such, falls results in predominant injuries to the elderly patients.

2.3.1.3 Quality of Life

The quality of life of patients after a fall continually decreases. According to Weeks and Roberto (2003), falls resulted in half of injury-related complications that hindered the elderly from undertaking normal daily activities. Notably, according to Taguchi et al (2016) elderly people with a good quality of life report a reduced number of falls. Furthermore, patients treated with fall related injuries such as hip fractures lose their independence in performing normal daily activities. Hartholt et al (2012) in a study conducted in a Dutch- population asserts that patients with hip fracture experienced problems in daily activity domains such as anxiety/depression (28%), mobility (90%), self-care (54%), and pain (69%). essentially, injury as a result of a fall is associated with a reduced quality of life and lack of independence.

2.4 Fall Detection Architecture

The overall system architecture for a fall detection system is similar in all the fall detection systems. Notably, the system comprises of different components that aid in the process of fall detection. The model architecture for the implemented systems comprises of similar architecture. The figure below depicts the basic architecture for fall detection systems:

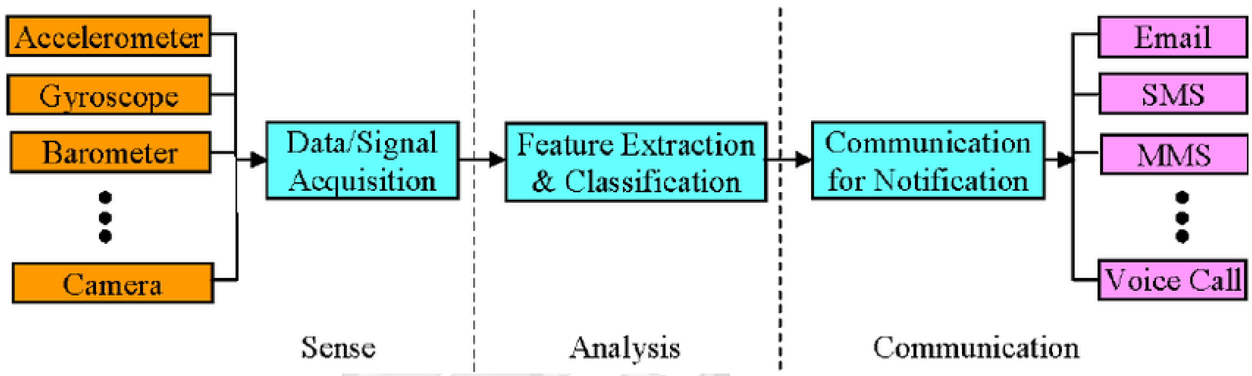

Figure 2.2: Fall Detection Architecture (Habib, 2014)

2.4.1 Sense

This component of the architecture focuses on sensing a fall occurrence. Essentially, this serves as the primary data collection point of a fall detection system. This component plays a vital role in the providing the correct input in form of data to the system to detect a fall. Notably, this component comprises of different sensors that may be used to retrieve the data. As highlighted in the figure above, the sensors may be an accelerometer, gyroscope, barometer and a camera. However, there are other types of sensors that may utilized in fall detection sensing. Notably, this serves as the classifying factor of fall detection systems. As such, there are wearable-device based systems, ambient-based systems, and vision-based systems.

2.4.2 Analysis

This component details the algorithms used to analyze the data provided by the sensors. The data captured by sensors ought to be analyzed to detect a fall from a normal daily activity. The algorithms implemented by fall detection systems can be classified in two distinct categories: Threshold-Based Algorithms and Machine-Learning Based Algorithms. Threshold based algorithms detect a fall if a pre-defined threshold value is attained by the analysis of a particular event. As such, the system extracts specific features from the data collected by the sensor and compares the values with the pre-defined values of the system. The system relies on the pre-defined values input by the system engineers in fall detection. On the other hand, machine-learning based systems use machine-learning algorithms to detect a fall. Essentially, with proper feature extraction from the input by the sensors, the algorithm classify an event as a fall or non-fall. In this systems, different machine learning algorithms have been implemented such as Hidden Markov Models and Convolution Neural Networks.

2.4.3 Communication

As highlighted in the diagram above, different modes of communication have been used to send the message from the fall detection system to concerned parties. The use of email, Short Message, Multimedia Messages and Voice Calls serve as the primary notification modes for fall detection systems.

2.5 Systems

Technology plays a vital role in the transformation of the healthcare industry. Most notably, the use of assistive or adaptive technology in the medical field enables the patients to attain a degree of independence. However, elderly patients still require adequate monitoring to inform 0medical practitioners in the case of emergencies. As such, various systems have been proposed to aid in fall detection for the elderly. Notably, the classifications of the systems depend on the technology used to provide the solution. According to Kaur and Kaur (2017), fall detection systems can be classified into wearable systems, vision-based systems and ambience sensor-based systems. Furthermore, each of the technologies utilizes different methods to detect falls. Wearable systems may be divided into inactivity based or posture-based systems. Additionally, vision-based systems fall in categories: Body shape change analysis, inactivity detection, and 3D head change analysis (Nizam, Mohd & Jamil, 2016). Lastly, ambience-based systems fall in two categories: Floor sensor detectors and posture. Essentially, each of the proposed systems presents a different methodology to detecting falls.

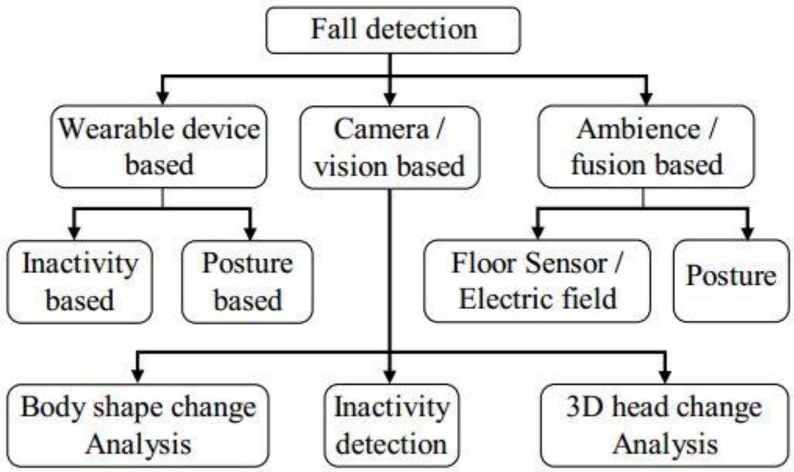

Figure 2.3: Fall Detection Classification (Nizam, 2016)

2.5.1 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 (2015) proposed a wearable device that uses an accelerometer to detect an elderly patient falling through an analysis of the acceleration. Essentially, the system calculates the total sum of the acceleration of the person. The total acceleration magnitude is calculated as shown below:

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

Equation 2.1: Total Sum Acceleration

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.

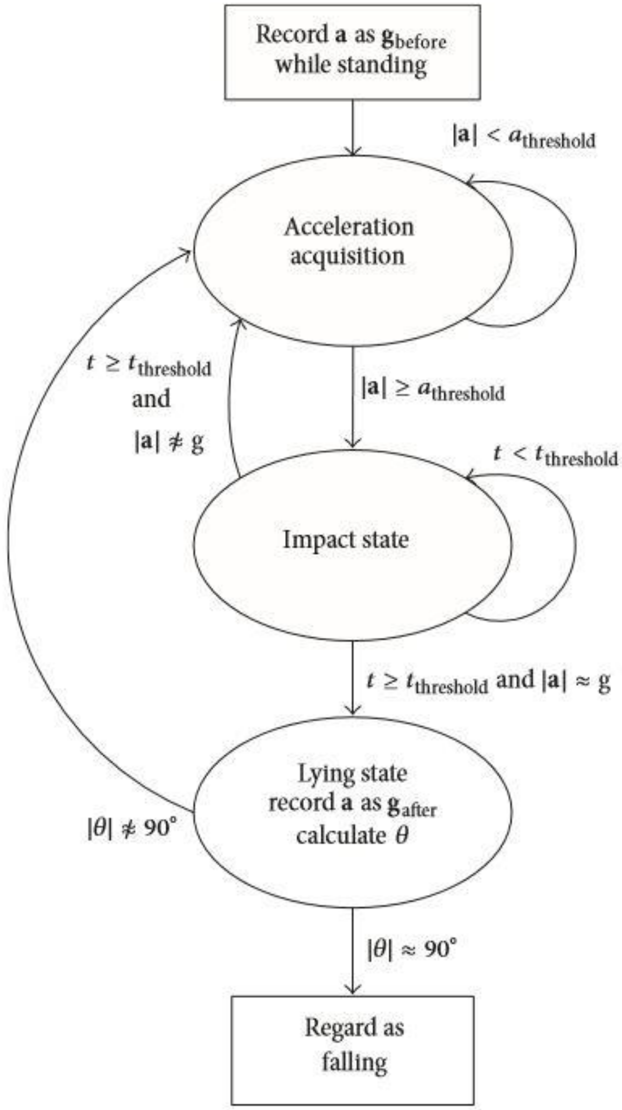

Figure 2.4: Wearable Fall Detection Algorithm (Wu, 2015)

Notably, Ngu et al. (2017) proposed a fall detection system using a smart watch coupled with a smart phone. The smart watch incorporates an accelerometer that detects the acceleration change of the person. The device then sends the selected feature values to a smartphone system that utilizes a machine learning approach to classify the event detected as a fall or non-fall. In the system, a Support Vector Machine classifies the event with the provision of selected feature values. The smart watch connects to the smartphone via a BLE link. Notably, the system encounters the challenge of the existence of a required connection with the smartphone. Therefore, without a connection to the smartphone the system fails to detect falls.

Gyroscopes aid to measure the angular velocity of an object. Notably, relying on the angular velocity does not sufficiently aid in fall detection. As such, the combination of a gyroscope and accelerometer presents improved accuracy and precision (Mao, Ma, He & Luo, 2017). Nari et al (2016) proposed a system placed on a person's waist with an accelerometer and gyroscope to detect a fall. The threshold based system aims at detecting a fall if predefined threshold values are measured by the sensors. Further, the system then communicates to a computer system through a Bluetooth link if a fall has been detected. The system achieved sensitivity and specificity levels of 90% and 86.7%. However, as highlighted earlier the system suffers from the challenge reliance on the connection to the computer.

Wibisono (2013) proposed a smartphone based system that utilizes the embedded gyroscopes and accelerometers for fall detection. Using a threshold-based approach the system aims at fall detection. Notably, some of the smartphone models are not equipped with the two sensors. As such, this presents a problem on the implementation of the system. Moreover, requiring each elderly patient to carry a smartphone poses a challenge since it may lead to discomfort. He, Bai, and Wang (2017) proposed a machine learning system based on the measurement values derived from attached sensors. A smartphone is used for computational purposes through the data values provided by the sensors. Essentially, with the integration of a Kalman filter the system reduces the noise presented by the sensors. Further, it uses a sliding window and the Bayes classifier to determine whether an event is a fall or an activity of daily living.

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.

2.5.2 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. The different applications of the system in detecting fall detection outline the different methodologies of the systems.

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) 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. In order to increase the accuracy of the system, the proposal included an inactivity window to prevent false positives.

Notably, Tabar (2006) proposed a smart home care system that utilizes accelerometers, image sensors and wireless network nodes to monitor the movements of a patient. The sensors provide the primary data in fall detection. Once a fall has been detected by the system, the image sensors activate to assert the occurrence of a fall.

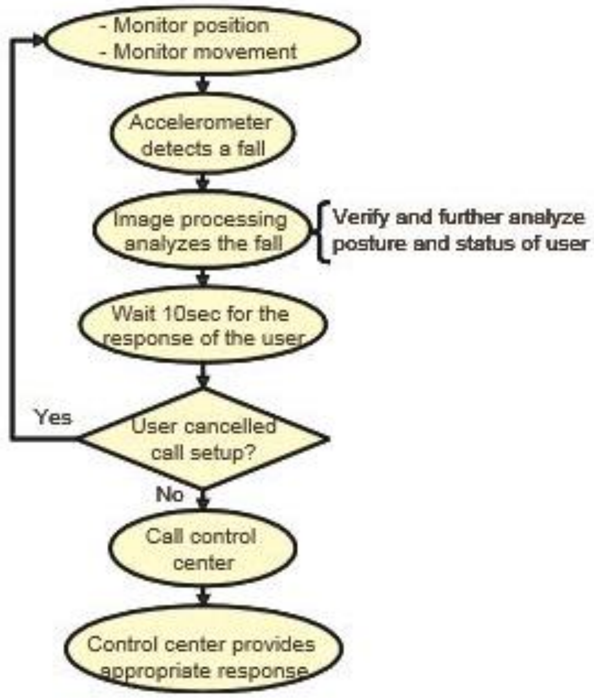

Figure 2.5: Flowchart of the smart home care system (Tabar, 2006)

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.

2.5.3 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:

2.5.3.1 Vision-Based Systems Architecture

i. 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.

ii. 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.

iii. Background Subtraction

Captured images contain irrelevant data related to the analysis of the data. Therefore, to focus on an individual on an image, background subtraction ensures that irrelevant 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.

2.5.3.2 Vision-Based System Classifications

Vision based systems can be categorized into three based on the mode of operation:

i. Inactivity

The systems presume that after a fall an individual is inactive. As such, the systems monitor a person to identify an inactivity. Jansen and Deklerck (2006) proposed a system that captures the 3D orientation of an individual with the use of a human silhouette. The system then observes a sequence of images detecting inactivity. Notably, it implements a context dependent inactivity detection that retrieves the location, time and duration of the inactivity. This seeks to reduce the occurrence of false positives. For instance, the system may detect that a subject is lying on the bed without an alert. Evidently, it uses a model that differentiates between normal activities of daily living and fall occurrences. As a result, this makes the model adaptive to the activities. Essentially, the system does not entirely present a good approach to fall detection since fall occurrences are random. Therefore, the use of context dependent inactivity method presents a challenge in real life random fall occurrences. Furthermore, as the researchers highlight, no validation of the system in a different setting from the simulated experiment.

Belshaw et al. (2011) proposed a system that combines body shape analysis with inactivity detection. In this approach, images of subjects submitted to the system are analyzed to detect a fall however certain areas in a room have been separated as inactivity sections. For instance, an individual on a bed may not be classified as a fall since the area around bed represents an area of inactivity. Notably, for fall detection based on the images, the researchers create a silhouette of the subject, extract features and classify the status as a fall or non-fall. With the use of neural network classifiers, the system achieves a true positive rate of 97% and a false positive rate of 5%.

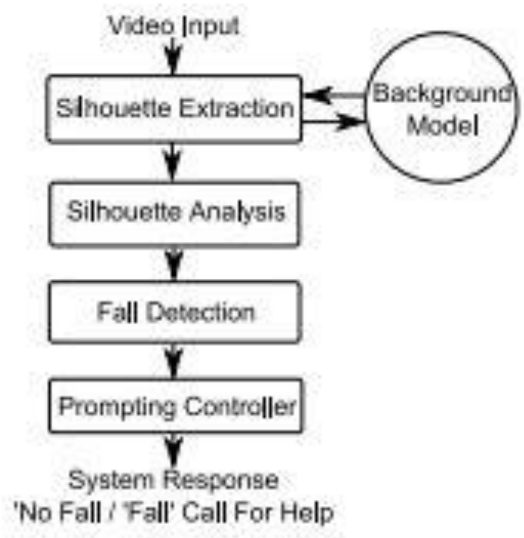

Figure 2.6: Inactivity Based (Belshaw, 2011)

ii. Head Motion Analysis

Head Motion Analysis approach in fall detection builds on the assumption that in the occurrence of a fall, the rate of change of vertical motion is higher than horizontal motion. Therefore, the technique analyses the rate of change of an individual's head to detect a fall. Rougier and Meunier (2006) proposed a 3D head tracking system that tracks the head of a person to detect a fall. Essentially, the method assumes that the head of an individual visibly appears in an image. Thus, using a particle filter the trajectory of the head can be derived. As such, with the tracking of the motion of the head, the threshold velocity for a fall detection is estimated which serves to identify a fall. The method suffers different disadvantages due to the incorrect identification of the head which may lead to incorrect fall detection. Furthermore, there have been less research on 3D head motion analysis affecting the level of accuracy of the approach.

iii. Shape Change Analysis

The approach assumes that for a fall event, the body shape of a human being changes in time. As such, this approach aims at detecting the change in shape of the human body to determine a fall or a daily activity. As highlighted earlier, machine-learning based and threshold based fall detection algorithms play an important role in accurate detection.

Rougier, Meunier, St-Arnaud and Rousseau (2007) proposed a threshold based shape change analysis system. The system implements Motion History Image to analyze the change in motion of an individual which combined with the analysis of the shape of the person determines a fall. With the use of optical flow, change in motion of an individual may be detected in a video sequence. This presents a challenge with the use of a real-time fall detection system. Therefore, the use of Motion History Image that provides a static representation of the change in motion of an object. Pixel intensity of a silhouette provides information of the recent position or motion of an object in a video. The figure 2.7 below, (a) represents an individual walking, (b) represents an individual falling, and (c) represents an individual undertaking a daily activity. Noteworthy, this presents an accurate approach that aids in inactivity detection.

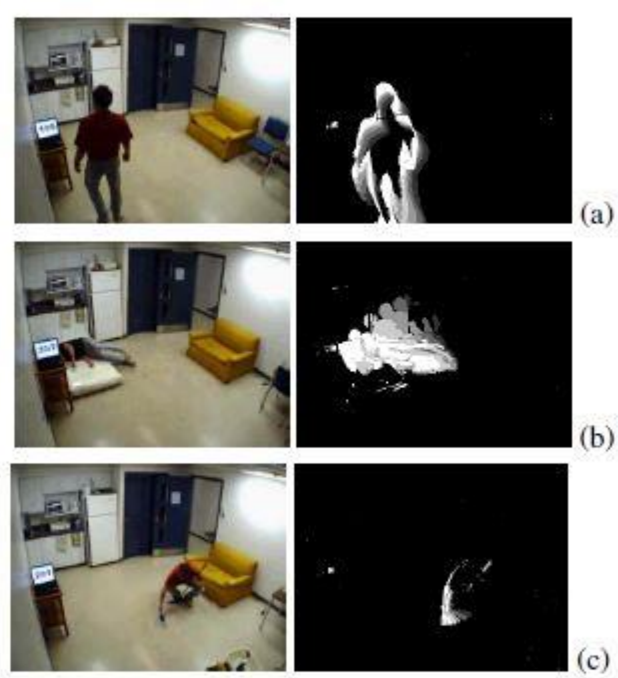

Figure 2.7: Motion History Image (Rougier, 2007)

Essentially, to increase the accuracy of their system they implement an inactivity check mechanism after the detection of a fall occurrence. Notably, this aids in reducing the number of false positives generated by the system. Evidently, the system requires high computing resources to undertake the process of background subtraction, motion history imaging and fall classification. Furthermore, the system relies on inactivity analysis to accurately identify a fall.

Thuc and Van Tuan (2014) proposed a system that implements background subtraction using the recursive Gaussian Mixture Model (GMM) approach. The system segments a human object in the frame and generates an ellipse model. As a result, this leads to accurate feature extraction. The features extracted from the silhouette frame ought to provide informative and discriminative data that clearly describe a fall occurrence. The features extracted from the silhouette include:

i. Current angle – This is the angle between the axis of the generated model and the horizontal axis of the present frame. Notably, this is derived by the equation 2.2 below:

$$\theta = \frac{1}{2} arctan \left\{ \frac{2 \sum i \sum j.x.y.I(i,j)}{\sum i \sum j.x^2.I(i,j) - \sum i \sum j.y^2.I(i,j)} \right\}$$

Equation 2.2: Current angle

i,*j*- represents the position of the pixel, $x = i - Ox$, $y = j - Oy$.

ii. Coefficient of motion- This represents the rate of motion of the individual in the current frame, t. This value depends on the motion of the previous 15 frames. Notably, the system implements Motion History Image (HMI) for motion comparison. Evidently, the coefficient of motion in a fall is relatively high than in other daily activities.

iii. Angle Deviation – This angle value represents the standard deviation of the previous 15 angles in the 15 frames.

iv. Eccentricity – Notably, this computes the variance of the ellipse from its circular shape using the equation 2.3 below:

$$e = \sqrt{1 - \frac{b^2}{a^2}}$$

Equation 2.3: Eccentricity

v. Centroid Deviation – The value represents the standard deviation of the centroid coordinates of the previous 15 images. Essentially, this value reduces exponentially in the occurrence of a fall.

Further, the extracted features are fed into a Hidden Markov Model (HMM) that classifies a fall and a non-fall. The system achieves an accuracy level of 100% on less challenging fall detection scenarios and a low of 82.35% on challenging fall detection scenarios. However, the average performance of the algorithm amounts to 87.38%. Notably, the system faces the challenge on the processing time required for video analysis of the algorithm to detect a fall. Essentially, this increases the amount of time required to detect and alert the case of a fall from 1 to 5 seconds. Evidently, this is as a result of the computation process in each of the frames such as background subtraction and feature extraction.

Núñez-Marcos, Azkune, and Arganda-Carreras (2017) proposed a convolution neural network fall detection system. The system uses optical flow images as input to the network aid in feature extraction for fall detection. Notably, to increase the efficiency and reduce processing time for the system, image processing steps such as background subtraction are omitted. Essentially, in earlier research, image processing led to the slow processing time of fall detection systems. Due to the limited number of datasets in fall detection, the adaptability of the CNN to be trained on different datasets is implemented on the system to increase the robustness and accuracy of the algorithm. Different datasets aid to train the model different image features that are important in fall detection. Further, optical flow images aid the network to identify different actions of an individual. Lastly, the use of transfer-learning and tuning the network layers focuses on fall detection.

2.5.3.3 Object Detection Algorithms

The analyzed algorithms above serve as the basis for a fall detection event in an image. Notably, the algorithms operate in scenarios of the presence of one major object (person) in an image. Therefore, the approaches above leverage on the presence of one object in each of the analyzed images. Essentially, in a real world environment, multiple people can appear in one image frame thereby complicating the use of the approaches above in vision-based approach. Object detection aims at identifying objects in an image and drawing a bounding box around the specific object. Different object detection algorithms have been developed with time which based on the Convolution Neural Network Architecture:

i. Region-Based Convolution Neural Network

Region-Based Convolution Neural Network (R-CNN) was proposed by Girshick et al (2014) to solve object detection region proposal issues. The architecture proposes the utilization of a method where a selective search algorithm extracts 2000 regions from an image. The selected regions from the image are warped into a square and fed to a convolution neural network that generates a feature vector. The extracted feature vector is the passed to a Support Vector Machine to classify the presence of an object within the region proposal. Additionally, the algorithm generates offset values that aid in the process of increasing the precision of the region proposal by adjusting the bounding box. The figure below shows the architecture of the algorithm:

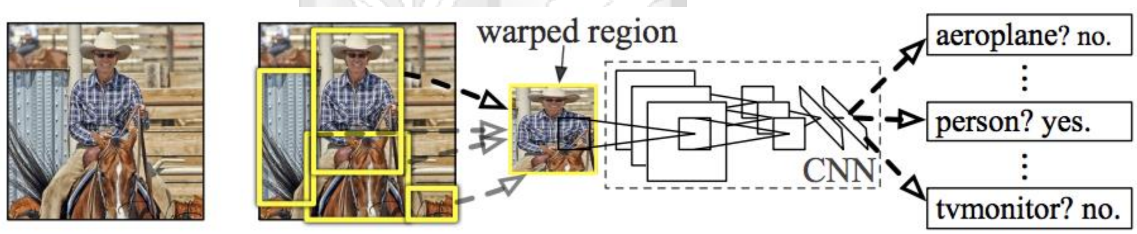

Figure 2.8: R-CNN (Girshick et al., 2014)

The first image depicts the input image captured through a camera. The next image shows the different proposed regions totaling to 2000. Additionally, the next image represents the use of Convolution Neural Networks to compute or derive necessary features from the proposed regions. Lastly, the network classifies if the region is specific to a particular class.

ii. Fast Region-Based Convolution Neural Network

Fast Region-Based Convolution Neural Network (Fast R-CNN) was proposed by Girshick (2015) as an advancement to the implemented Region Convolution Neural

Network. The network takes an image as an input and specified object proposals. This enables the network to process the image through convolution neural networks and max pooling layers to provide convolution feature maps as an output. The convolution feature map enables the network to identify region proposals which are warped into squares and with the use of a Region of Interest (RoI) pooling layer the proposal regions are reshaped to be fed to a fully connected layer. To predict the specific class, a softmax layer is implemented from the RoI feature vector. The figure below shows the architecture of Fast RCNN:

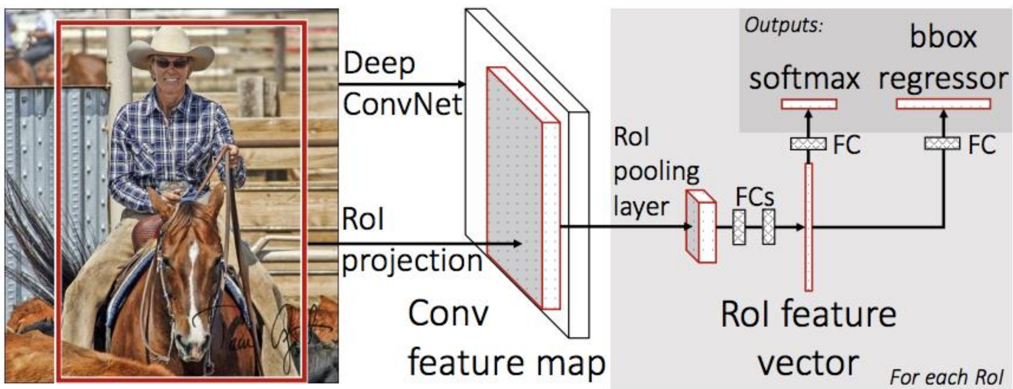

Figure 2.9: Fast RCNN (Girshick, 2015)

The performance of the network improved as the process of generating the region proposals was replaced with the use of the convolution operation. Essentially, this leads to the reduction in processing time for each of the images while training and also while testing. Though the network performs faster in processing time, the use of region proposals for the images has significant impact on its performance time.

iii. Faster Region-Based Convolution Neural Network

R-CNN and Fast R-CNN leverage on the implementation of the selective search algorithm to identify the region proposals. However, the using selective search leads to a slow process leading to time wastage thus affecting the performance of the network. To curb the slow process and time wastage, Ren et al. (2015) proposed the use of a region proposal network to predict the region proposals. Essentially, the region proposal network comprises of a fully convolution neural network that is able to predict the object bounds and the score of the object at each position. Further, the region proposals are then passed to a Fast R-CNN detector which uses the proposed regions to detect the specified objects. The diagram below explains the architecture of the system:

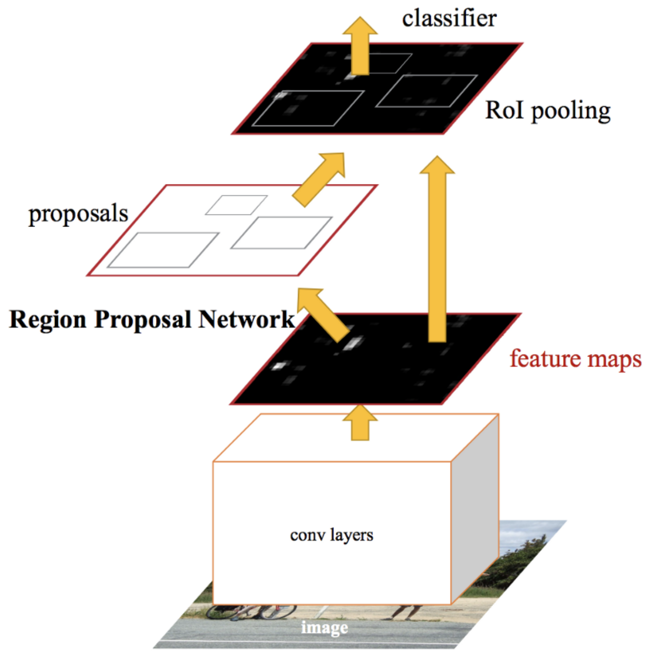

Figure 2.10: Faster R-CNN (Ren et al. 2015)

Notably, the use of the region proposal network increases the speed of execution of the algorithm to aid in its use in real-time object detection. The above algorithm plays a significant role in identifying different objects within an image and the local position within the image. As such, the algorithm can classify different classes of objects in an image. This enables proper classification of object classes however does not clearly provide intra-class classification that challenges multi-person fall detection. Therefore, to undertake the multi-person fall detection the algorithm below is chosen.

iv. Mask Region-Based Convolution Neural Network

Mask R-CNN builds on the implementation of Faster R-CNN to create object masks on each of the identified object in the image. The algorithm was proposed by the proponents of the Faster RCNN model (He, Gkioxari, Dollár, & Girshick, 2017). This enables the model to accurately aid in 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 acquires high accuracy levels on the first step then the system will achieve high accuracy levels in the second step. As earlier stated the model utilizes the achievements of Faster R-CNN to properly undertake semantic segmentation. Therefore, the algorithm adds a branch of segmentation masks on each of the Region of Interest (RoI) which runs in parallel with the classification branch. Essentially, this helps the algorithm to easily create a pixel-wise mask on the different instances of each class. As such, the algorithm will be implemented to aid in fall detection in a multi-person environment due to its capabilities as highlighted above.

2.6 Conceptual Framework

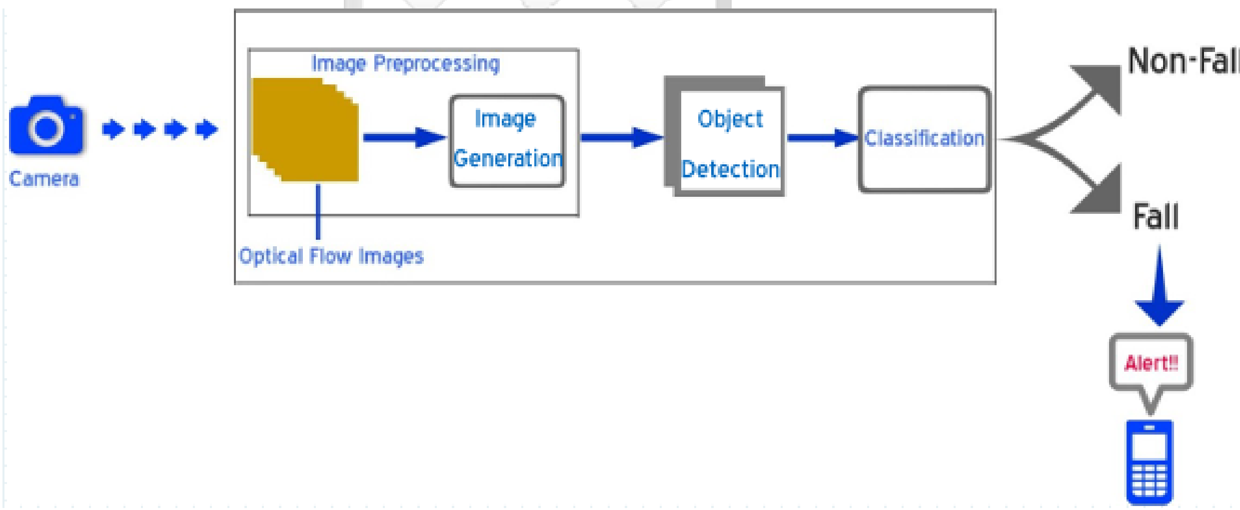

Figure 2.11: Conceptual Framework

The proposed conceptual framework for a multi-person fall detection system is as illustrated above. It details the input as a series of images from a camera with more than one object or person present in the image, the model extracts images while analyzing each of the images. Firstly, the system receives an optical flow of frames from the camera, for each frame the system generates an image. Further, once the images have been generated the system runs object detection on each of the image to determine the existence of an object mostly a person. The system then classifies whether the person has fallen down, if it detects a fall the system then sends a notification message to the registered personnel else if no fall detected the system does not provide any notification message.

Chapter Three: Research Methodology

3.1 Introduction

Research methodology outlines the methodology used in conducting the research. Essentially, it outlines the procedure used in the collection, the information on the variety of participants and the mode of inclusion in the study. Additionally, the research design of the study is outlined with the provision of the specific explanation of the selected choice. Notably, it highlights the location, target population of the study and the sample population used for the study. Furthermore, it describes the system methodology used in the research for the paper explaining the pros of the methodology over other existent methodologies. Lastly, the ethical considerations for the research are outlined in the content.

3.2 Research Design

The research was experimental in nature as it aimed at detecting a fall scenario in a multi-person environment within a home-based care scenario. Essentially, the developed system utilizes images retrieved from the camera to classify whether a specified image contains a person who has fallen. Therefore, experimental data was generated to help in the development of the model for classifying the fall scenarios.

The research paradigm used for the study was quantitative as the experimental data retrieved from the input devices that is the sensors is represented in numbers thereby quantified to present the output. According to Creswell and Clark (2007), since the research presents a systematic empirical investigation of observable phenomena, thereby requiring the utilization of quantitative data. Creswell and Clark (2007) asserts that quantitative research is based on the measurement of quantities. The research requires accurate results due to its sensitivity in regards to time and the output therefore, the need to quantify the data.

3.3 Location of the Study

The research was undertaken in Strathmore University since the institution provided the needed infrastructure in regards to the required equipment that was used for the study. Furthermore, due to the nature of the research, the institution provided the required number of participants required for the study. Lastly, it provided an enabling environment for the required simulations to be conducted necessary for the research.

3.4 Population and Sampling

3.4.1 Population

The target population for the study was elderly patients. Globally, in reference to the United Nations ageing population report, there are 901 million elderly people in the world. As the statistics dictate the number is expected to increase to 1.4 billion by 2030. In Kenya, according to the National Bureau of Statistics, as of 2009, the number of elderly people is 1.9 million. Arguably, this presents about 4% of the total population. However, as projected, life expectancy in the world is increasing, therefore, the number of elderly people in the country is expected to increase. Notably, there is no proper documentation to state the number of elderly patients in different home-based care centers though there exists a number of the centers in different parts of the country.

3.4.2 Sampling

The research focused on probability sampling to prevent bias of the sample to suit the needs of the researcher. The target population for the research is elderly people however, normal people simulated fall events to retrieve the required simulation images. Mostly, this is due to the fragility of exposing the elderly patients to the risk exposure of falls. The research was undertaken with four participants in different postures generating the required images. The process of participating in the study was voluntary and did not involve the provision of any reimbursements.

3.5 Data Collection

The primary data collection method for the study was through capturing images of fall scenarios with the utilization of a camera. The researcher captured 139 images used in undertaking the research. The captured images were taken in Strathmore University and contained images of participants in two postures: in fallen posture and standing posture. Essentially, the camera served as the primary sensor in the gathering of information for the study. The cameras recorded different simulations of a fall incident providing the optical flow of images for the development of the model.

3.6 Model Development

The process of model development entailed different steps aimed at providing the suitable model for the system. Notably, the steps can be outlined as shown below:

i. Image Preprocessing and Data Splitting.
ii. Development of the model.
iii. Validation of the model.

3.6.1 Image Preprocessing and Data Splitting

The captured images were converted from the default bitmap image file to jpg 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 73% training and 27% 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.

3.6.2 Development of the Model

The input images served as the basis for the generation of the fall model. The Mask R-CNN model resized the images to a minimum of 800 by 800 and a maximum of 1024 by 1024 (Abdulla, 2017). This ensured the proper generation of the region proposals for each of the image. Notably, the architecture generates 200 Region of Interest for each of the images provided. As such, this enables the network to process each of the images faster for the training and validation processes. Additionally, to reduce overfitting, the network utilizes a weight decay of 0.0001. The model is trained on 10 epochs each with 100 steps.

3.6.3 Validation of the model

The generated model was tested against the validation dataset. Essentially, the metrics used to test the model include: mean Average Precision and Loss Function. The metrics serve as the best indicator of the performance of the model on the provided task. The model outputs a mask with the fall classification.

3.7 System Development Methodology

3.7.1 Agile System Development

Agile software development incorporates iterative and incremental methodology. The methodology was implemented in a manner that promoted the continuous development of the system with new developments. The development methodology supported the design of the system since it required continuous testing of the fall detection system. Essentially, due to the intensity of the information provided by the system, agile software development provided the opportunity to redesign and test the developed system.

3.8 System Design

The system architecture comprises of hardware components and software components. The hardware components implemented for the system include the camera and a processor. The software components of the system include the fall detection algorithm used to classify a fall. The camera captures the image needed for analysis by the developed model. Essentially, if a fall is detected the system sends a notification message to the registered caretaker. Additionally, the system implemented a functional database to store the information of the patient and the emergency response information.

3.9 Research Quality, Reliability and Validity

Quality research entails the scientific process that encompasses all aspects of study design; mostly related to the match between the questions and methods, choosing of participants, measurement of outcomes, and protection from inferential errors or bias (Boaz & Ashby, 2003; Lohr, 2004; Towne & Shavelson, 2002). This research followed the required quality standards by ensuring proper procedure was used in the collection of data, accurate precision in the citation of external sources, and the results obtained presented without manipulation. The collected data was representative of the study and the use of corrective mechanisms applied to enhance the quality of research. As highlighted earlier, each of the derived models was tested against standard metric to ascertain its performance.

3.10 Ethical Considerations

The research entailed the capturing of participant's images to use for data collection, therefore, the researcher sought permission from relevant authorities to collect the data. Further, the researcher requested volunteers to undertake in the study thereby ensuring each of the participants had the freedom to exit the research at any stage.

Additionally, the data collected for the research was only used for the specified research thus preventing access to the data by any external parties. With the adherence to above procedure, the research attains required ethical considerations.

Chapter Four: System Analysis and Design

4.1 Introduction

System analysis and design plays a critical role in the development process of systems. Essentially, this enables the system analyst to have a proper structure on the way to develop the required system. Furthermore, it diagrammatically represents the system enabling a better understanding by stakeholders. The analysis and design of the systems conform to the world-wide accepted UML diagram notation clearly depicting the environment of the system with all the entities interacting within the system.

4.2 Requirement Analysis

Essentially, to develop the proposed system, requirements gathered were placed in two categories: Functional requirements and Non-functional requirements.

4.2.1 Functional Requirements

(i) The system should accept a series of images from the camera.

(ii) The system should transform each of the images and extract relevant features.

(iii) The system should identify the posture of the people in the image.

(iv) The system should provide an output whether any of the persons in the image has fallen or not.

(v) The system should send a notification to the caretaker if a person has fallen.

4.2.2 Non-Functional Requirements

4.2.2.1 Supportability Requirements

The system should be able to be supported with a wide range of input cameras that can easily interface to provide the desired output. Essentially, this promotes the usability of the system in the real-world environment.

4.2.2.2 Reliability Requirements

The system should consistently provide the same result provided with the same input data. Essentially, this ensures a predicted result which conforms to the predicted output hence easier configuration of the system. Additionally, the system needs to accurately identify falls and reduce false classifications.

4.2.2.3 Generalization Metrics

The system should generalize with new instances of input data to increase the accuracy of fall detection. Essentially, this ensures that it low failure fall detection rate which may affect the accuracy and reliability of the system.

4.2.2.4 Ease of reconfiguration

The system implements a region-based convolution neural network which requires reconfiguration of the weights to achieve a high level of accuracy classification. As such, the system should have an easier mode which the weights of the model can be reconfigured to achieve the required results.

4.2.2.5 Security Requirements

The system captures images of people in a specific room which intrudes personal privacy. As such, the system needs to implement strong security measures to ensure its integrity and protect privacy. Furthermore, the alteration of the system configurations require authorized personnel to increase the security of the system.

4.3 System Design and Architecture

4.3.1 System Model Architecture

The system uses a camera as the only source of input data. A camera will capture a series of images of the person with a defined period of time. The series of images will be passed to the generated model for fall classification. Essentially, to achieve high levels of accuracy of fall classification, the generated model is trained and tested on fall simulation data as shown in the figure below:

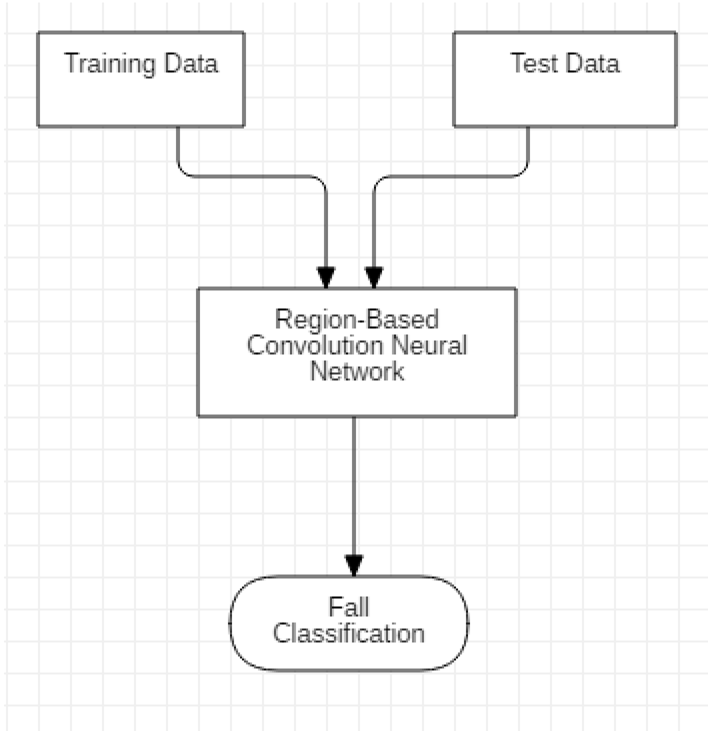

Figure 4.1: System Model Architecture

4.3.2 *Use Case Diagram*

A use case diagram in UML diagram represents the users interacting with the system. Notably, the proposed system consists of three main actors: Administrator, Developer, and Caretaker. The administrator will undertake the task of entering mapping information of the different locations of cameras and caretaker details. This will enable the system to send alerts to users in case of a fall. Additionally, the developer of the system continually updates the system model used in fall classification.

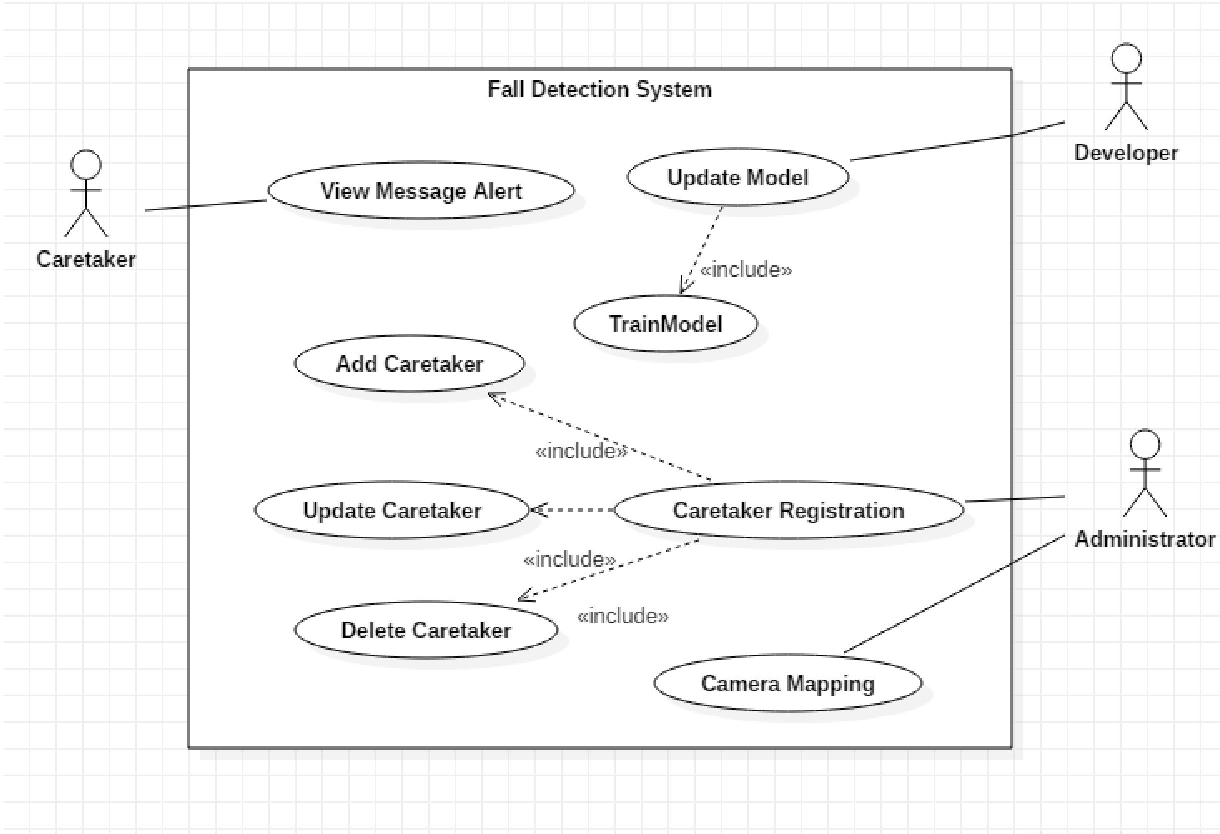

Figure 4.2: Use Case Diagram

The table below represent the view message alert use case depicted above:

Table 4.1: View Message Alert

Use Case : View Message Alert	
Primary Actors : Caretaker	
Precondition : Fall detected	
Post-condition : Series of image analysis	
Main Success Scenario	
Actor responsibility	**System Responsibility**

	1. The system sends an alert message to the specified caretaker.
2. The user views the alert message	

The caretaker registration, update model, camera mapping tables have been placed in Appendix A as Table 8.1, 8.2 and 8.3 respectively. Each of the tables provides the main success scenario with the responsibilities of each of the actors in the use case. Further, Table 8.1 and 8.2 provide additional alternative scenarios in case part of the main success scenario fails. This provides different scenarios in case the system encounters errors.

4.3.3 Data Flow Diagram

A data flow diagram depicts the processes and entities in a system outlining how data flows from each of the entities to the processes. Additionally, it captures the storage of data from the processes. Essentially, it primarily shows the flow of data in the system aiding in a better understanding of the system. In this system, the diagram will outline entities such as the users of the system.

4.3.3.1 Context Diagram

Context Diagrams represent high level data flow between processes and entities in a system. This gives a brief overview of the data flow in the system. The diagram below depicts the context diagram of the desired system.

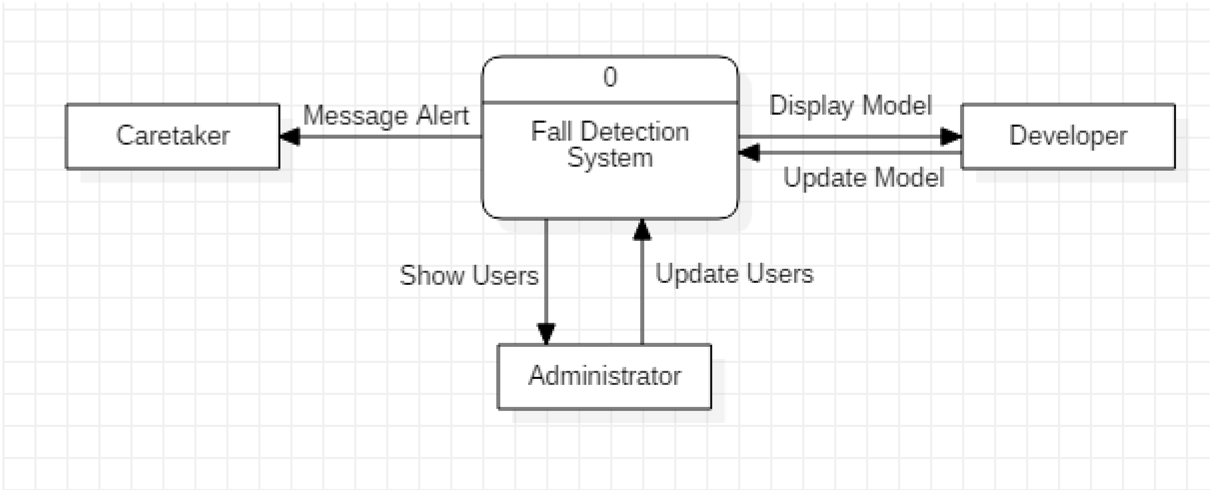

Figure 4.3: Context Diagram

4.3.3.2 Level 0 Diagram

Generally, to clearly represent the specific processes, entities and data stores in a system, a level 0 diagram is drawn to support the context diagram information. This enables a clear understanding of the data flow process.

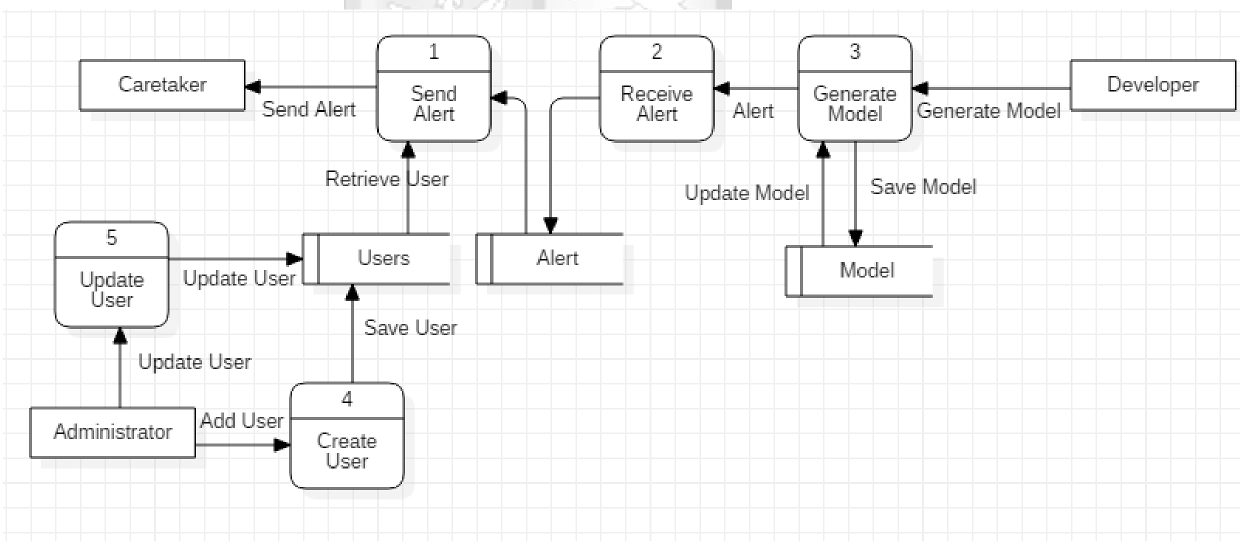

Figure 4.4: Level 0 Diagram

4.3.4 Sequence Diagram

Sequence diagrams outline the interaction between the objects in the system. Essentially, the diagram clearly shows in sequence how a system achieves a specific objective. The diagram shows the steps undertaken through the fall detection system. The developer initializes a model that learns from an input of a series of images. The images are first pre-processed to conform to the required specifications. The model is saved to the memory and provided with a new series of test images to classify.

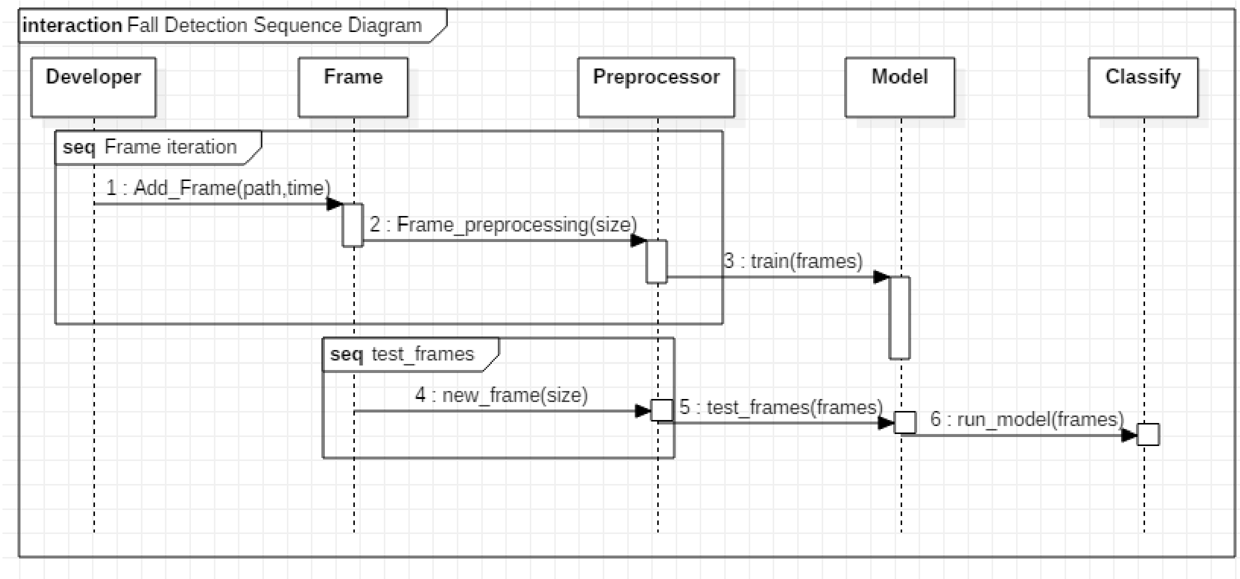

Figure 4.5: Sequence Diagram

4.3.5 Class Diagram

Class diagrams provide an overview of the structure of the system, that is, the classes in the system and their interactions. This ensures a better understating of the system by the developers.

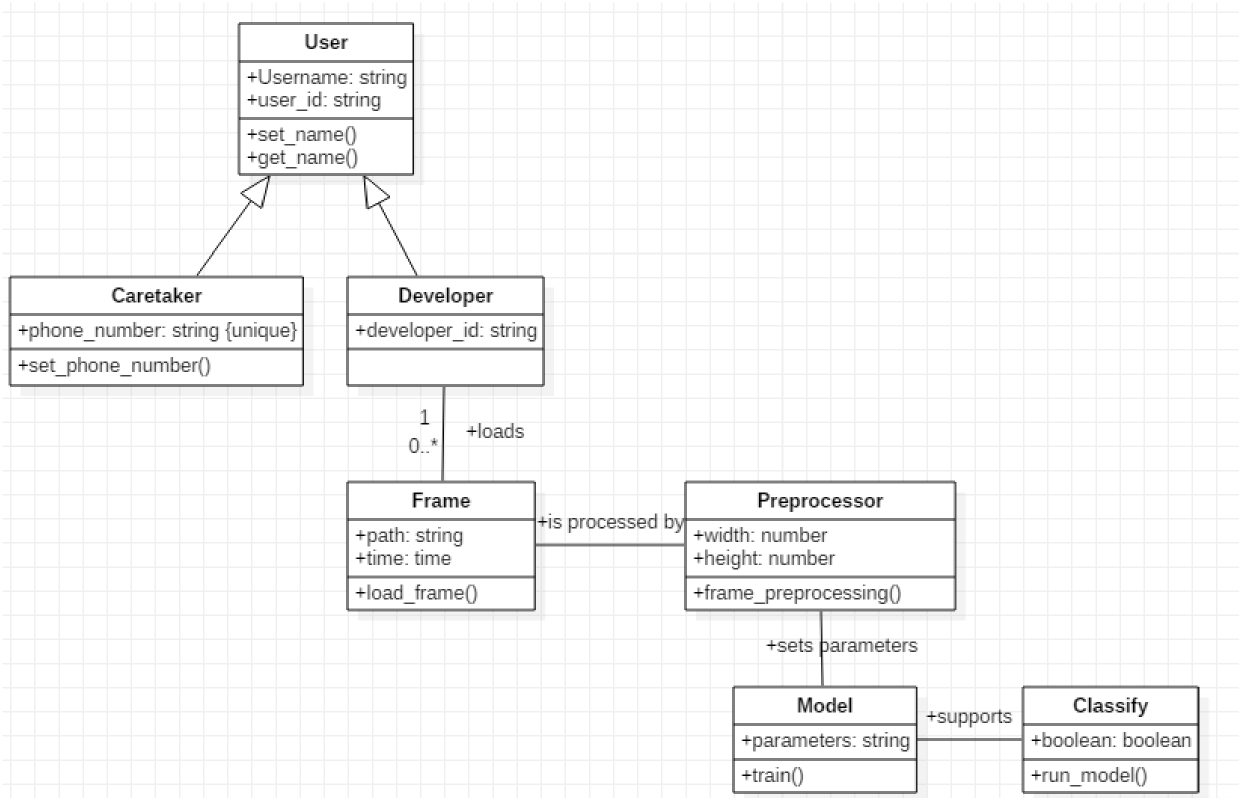

Figure 4.6: Class Diagram

4.3.6 Database Schema

A database schema shows the different entities in a system and their relationships that affects the association between them. Furthermore, the schema highlights the constraints placed on the system. The diagram below shows the database schema for the system.

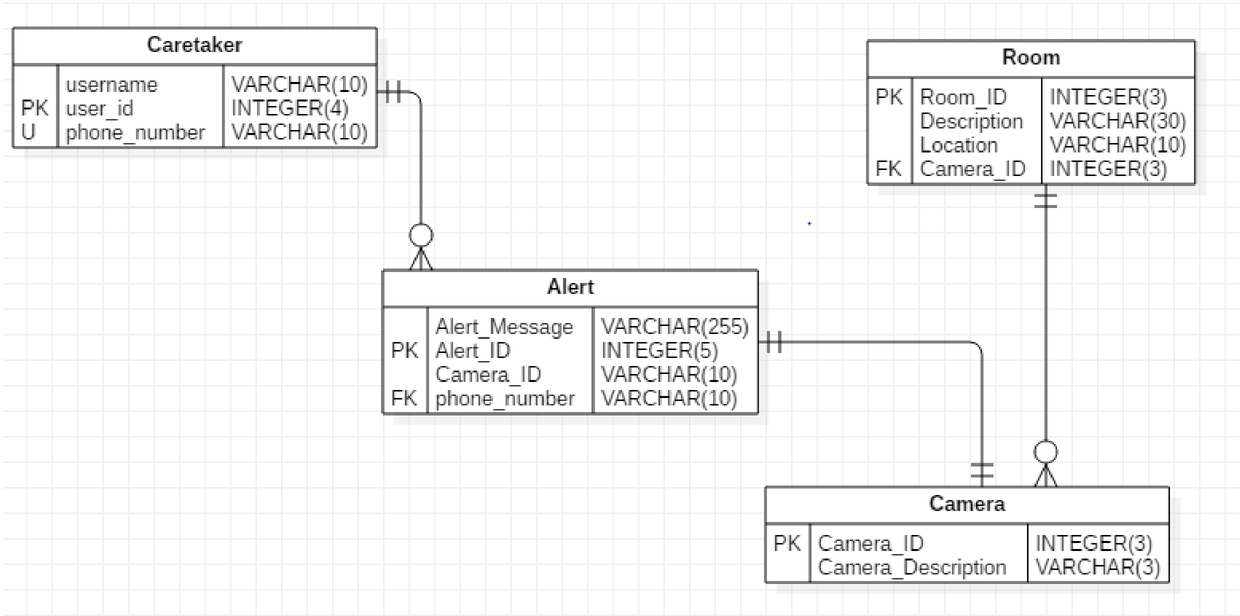

Figure 4.7: Database schema

Chapter Five: Implementation and Testing

5.1 Introduction

Implementation and testing incorporates a series of steps that determine the nature of the delivered system. The implementation stage of the system included the preparation of the captured images, different tools used to generate the model, various steps undertaken in programming and the validation of the model. The preparation of the images entailed the conversion of the images to a desired format required for the model and subsequent annotation of the images to match specified classes. Further, with the training data annotated, a desired model is built for the fall detection system. The fall detection system incorporates the development of a Short Message Service notification system supported by a MySQL database.

5.2 Hardware and Software Environment

The development of the model was undertaken on Google Cloud Platform. A cloud virtual machine with the Ubuntu Operating System was chosen as it provided the required computing resources. The default programming language for the development of the model was Python due to the availability of the required libraries. Essentially, different computer vision and specified machine learning libraries were used: OpenCV, Tensorflow, Keras, PIL and Numpy. They provided different functionalities required for the system development. The table below highlights the different specifications used for the model:

Table 5.1: Hardware and Software Environment

Software	**Specified Library**	**Version**
Python (3.5)	Tensorflow	1.13.1
	Numpy	1.16.2
	OpenCV	4.0.0
	Keras	2.2.4
Hardware	**Details**	

Google Cloud	RAM	13GB
	vCPU	2 (Intel Broadwell)
	GPU	1 (NVDIA Tesla K80)

5.3 Dataset Preprocessing

A total of 276 images were captured from different participants with the use of the Microsoft Kinect v2. Notably, a total of 139 images were deleted as they were not representative of the aim of the study. Therefore, a total of 137 images were used for the study. Additionally, the images were then divided into train and validation sets as 100 and 37 respectively. Notably, the Microsoft Kinect v2 captures the images in bitmap format which is not compatible with the required format, as such, a python function was used that converted the images from bitmap to jpg. The image conversion process did not alter the aspect ratio of the image but greatly reduced the size of the images. The python function is listed below:

```
def converter(target_directory,target,final):
    for file in listdir(target_directory):
    filename, extension = splitext(file)
    #print(target_directory)
    try:
        im = Image.open(target_directory + '/'+ filename + extension)
        im.save(final + '/'+ filename + target)
        im.close()
    except OSError:
        print('Cannot convert %s' % file)
```

The converted images were then split in two different locations for the image annotations. One set of images was annotated with the use of one class, that is, to identify only a fall in an image. Furthermore, the second set of images was annotated with two separate classes, that is, a fall class and a standing class. This was aimed at providing a comparative study between the two fall classification scenarios.

Table 5.2: One Class Configuration Dataset

Dataset	**Dimension**	**Number of Classes**
Training	(100,1920,1080)	1
Validation	(37,1920,1080)	1

Table 5.3: Two Class Configuration Dataset

Dataset	**Dimension**	**Number of Classes**
Training	(100,1920,1080)	2
Validation	(37,1920,1080)	2

Image annotation was achieved using the VGG Image Annotation tool (version 1.0.6) which generated a JSON file with the desired classes. Each of the images in the dataset were annotated depicting the different classes present in the images. Notably, as highlighted earlier, two different sets of classes were used for evaluation purposes. The images below represent the different annotations used for the model.

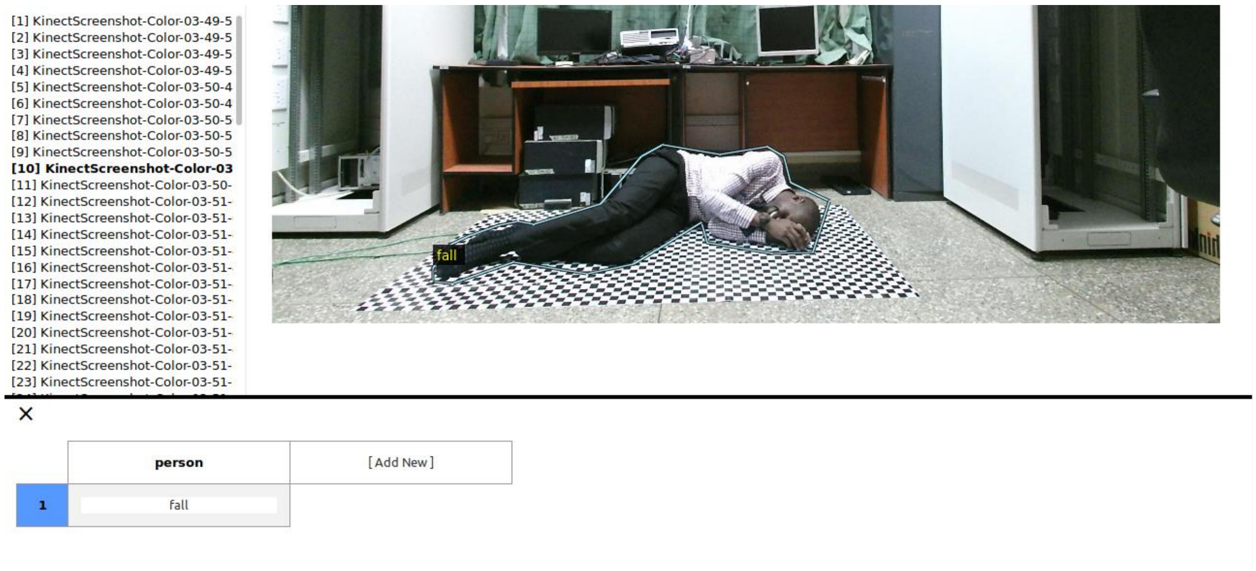

Figure 5.1: One Class Image Configuration

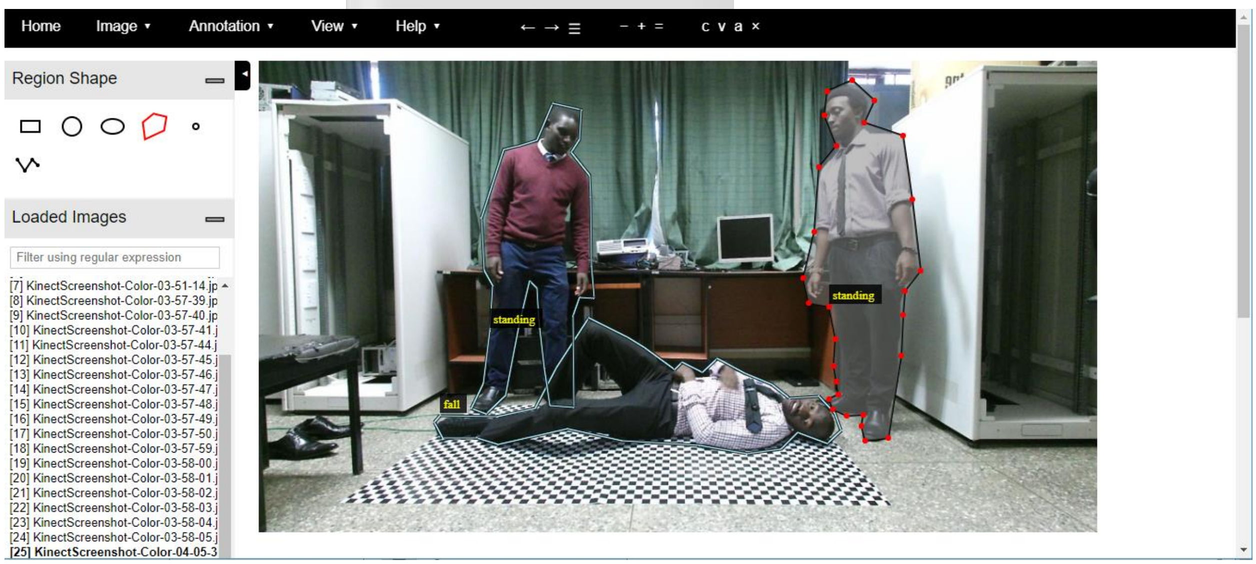

Figure 5.2: Two Class Configuration

The generated JSON file contains a unique identifier of the image with the corresponding details. Notably, in each of the images, the coordinates of each of the classes is stored for the class identification. The output of the JSON file is represented below for the two class configuration:

Figure 5.3: Two Class Configuration JSON Output

5.4 Mask RCNN Model

Mask RCNN is classified as an instance segmentation model that identifies different instances in an image. The researcher did not configure the hidden layers of the model nor the activation functions used in the presented Mask RCNN. Essentially, this was aimed at maintain consistency in the different configurations for the study as discussed thereafter. The developed model was trained on two different configurations that provided different results.

5.4.1 One Class Configuration

Notably, in the first configuration, the labelled data was annotated with only one class label, that is, the fall class. The Mask RCNN used in this instance recognized the background as a separate class. The aim was to only correctly identify a fall scenario in one picture. The configuration code used to define the different classes used for the model defined different parameters: Firstly, as described above, the number of classes present in each image annotation. Secondly, the specifications of the type of processor used for the research. Additionally, the number of steps per each epoch which dictates the number of steps before a training epoch is declared finished. Lastly, the detection minimum confidence which dictates the confidence level of the detections in the images. The sample code below provides the specifications used for the first configuration:

```
class BasicConfig(Config):
    NAME = "fall"
    IMAGES_PER_GPU = 1
    NUM_CLASSES = 1 + 1
    STEPS_PER_EPOCH = 100
    DETECTION_MIN_CONFIDENCE = 0.9
```

The specified images were then loaded using a specified function described below that ensured the derivation of the annotations of the x and y values of fall identified region. The function accepts the dataset directory to retrieve the required images and the JSON file generated from the VGG Annotator Tool. The function loads the JSON file to access the different annotations of regions with the specified class fall region. This serves as the input to the Mask RCNN model for the specified class.

```
    def load_custom(self, dataset_dir, subset)
        self.add_class("fall", 1, "fall")
        assert subset in ["train", "val"]
        dataset_dir = os.path.join(dataset_dir, subset)
        fallannotations = json.load(open(os.path.join(dataset_dir, "via_region_data.json")))
        annotations = list(fallannotations.values())
        annotations = [a for a in annotations if a['regions']]
        for a in annotations:
            polygons = [r['shape_attributes'] for r in a['regions'].values()]
            image_path = os.path.join(dataset_dir, a['filename'])
            image = skimage.io.imread(image_path)
            height, width = image.shape[:2]
            self.add_image(
```

```
"fall",
image_id=a['filename'],
path=image_path,
width=width, height=height,
polygons=polygons)
```

Essentially, this ensured the different identified regions encompassing a fall region have been loaded to the model. The model was trained for only 10 epochs for the network heads. Due to the limitation of the data used in the development of the model, transfer learning approach was used to increase the accuracy of the model. Therefore, weights of the COCO dataset were loaded to the model to enhance the performance of the model.

5.4.2 Two Class Configuration

The class configuration represented the real world model where an image consists of more than one instance of a specified class. Notably, the images captured comprised of different people in specified pose positions. Mostly, the distinct positions comprised of a fall position and a standing position. As such, the positions were presented as two distinct classes of a person. This served as the basis of the distinct instance segmentation of the position of two people in an image. The configuration code used for the two class configuration was updated to represent the different classes present in an image. As stated earlier, the background of an image is added as a class, therefore, the total number of classes represented was 3. The other parameters such as the number of steps and number of images per GPU were not altered. The sample code below represents the scenario:

```
class BasicConfig(Config):
  NAME = "fall"
  IMAGES_PER_GPU = 1
  NUM_CLASSES = 1 + 2
  STEPS_PER_EPOCH = 10
  DETECTION_MIN_CONFIDENCE = 0.9
```

The specified images were then loaded using a specified function described below that specified the different classes present in the image and the representative x and y regions. The function is similar to the one class configuration only that different number of classes and a specification of additional variables to cater for the different classes.

```
def load_custom(self, dataset_dir, subset):
    self.add_class("fall", 1, "fall")
    self.add_class("standing", 2, "standing")
    assert subset in ["train", "val"]
    dataset_dir = os.path.join(dataset_dir, subset)
    fallannotations = json.load(open(os.path.join(dataset_dir, "via_region_data.json")))
    annotations = list(fallannotations.values())
    annotations = [a for a in annotations if a['regions']]
    for a in annotations:
        polygons = [r['shape_attributes'] for r in a['regions'].values()]
        names = [s['region_attributes'] for s in a['regions'].values()]
        image_path = os.path.join(dataset_dir, a['filename'])
        image = skimage.io.imread(image_path)
        height, width = image.shape[:2]
        self.add_image(
            "fall",
            image_id=a['filename'],
            path=image_path,
            width=width, height=height,
            polygons=polygons,
            names=names)
```

The two class configuration model was trained on 10 epochs with 100 steps per epoch. As such to ensure consistency for proper evaluation of the two models, transfer learning was implemented using the COCO dataset weights.

5.5 System Implementation

5.5.1 Storing System Details

The system works by providing fall detection notification to the registered users of the specified room. Therefore, to ensure proper notification of users, a MySQL database was created to easily manage the notification process. The figure below shows a snapshot of the tables in the database:

```
mysql> SHOW TABLES;
+------------------+
| Tables_in_Thesis |
+------------------+
| Alert            |
| Camera           |
| Caretaker        |
| Room             |
+------------------+
4 rows in set (0.00 sec)
```

Figure 5.4: Database Details

5.5.2 System SMS Notification

The system provides a notification to the caretaker through a Short Message Service to the caretaker's registered phone number. Essentially, this serves as the default notification method to inform the caretaker of a detected fall incident. Notably, to quicken the response time, the system clearly stipulates the specified room that the fall incident has occurred and a description of the room details. The information is retrieved from the database aiding in the easier identification of a fall incident. Additionally, the system only sends a notification only when a fall incident is identified. The figure below depicts a sample notification message sent to a registered caretaker:

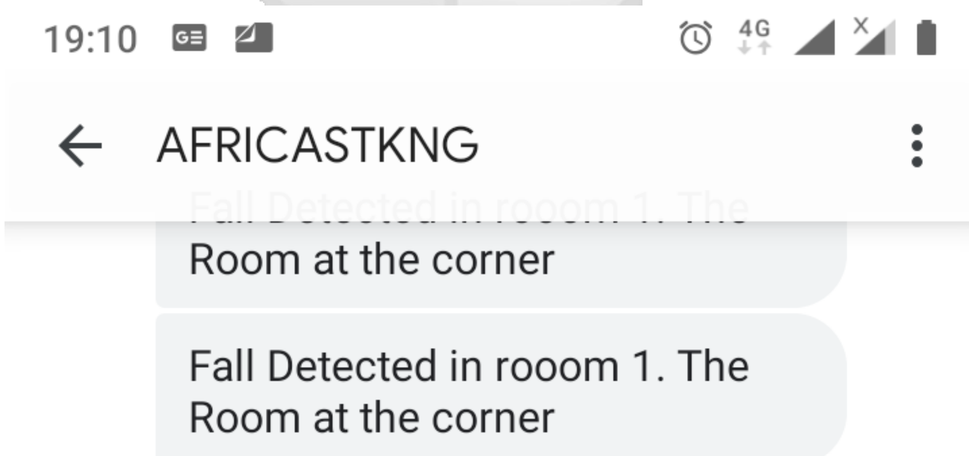

Figure 5.5: SMS Notification Alert

5.6 System Testing

The system aims at providing real-time detection of a fall in a multi-object environment. With the use of a video camera, real time images are captured and analyzed by the generated model for fall detection. Notably, for the testing of the system, a video of a falling person is used as the input to the system for the generation of images. Essentially, this serves as a simulation of a real time fall event. The generated model receives images as input for fall classification, therefore, each of the captured frame in the real-time video needs conversion to an image format. The code below aids in capturing frames and creating images required for the model:

```
def extractImages(pathOut):
  count = 0
  vidcap = cv2.VideoCapture(0)
  success,image = vidcap.read()
  success = True
  while success:
    vidcap.set(cv2.CAP_PROP_POS_MSEC,(count*1000))
    success,image = vidcap.read()
    print ('Read a new frame: ', success)
    cv2.imwrite(pathOut + "/" + "frame%d.jpg" % count, image)
    count = count + 1
    if count==22:
        success=False
  vidcap.release()
```

Furthermore, to evaluate the system appropriately, a test case generation plays an important role to depict the performance of the system. As such, the test cases below seeks to validate the system's functionality in comparison with the proposed requirements.

Table 5.4: System Test Cases

Test Case	Importance Level	Results
Does the system capture real-time images?	High	The system captures real-time images with the use of a camera.
Does the system classify the objects in the image?	High	The system analyzes the images and classifies fall scenarios.
Does the system provide for the alteration or update of user's information?	High	With the use of a database, user's details can be easily updated.
Does the system provide notifications to the users?	High	The system provides a notification message in case of a detected fall incident.

Chapter Six: Discussions

6.1 Introduction

The study aims at developing a multi-person fall detection system with a SMS notification system. This assumes the generation of an efficient fall detection model to correctly identify fall events. Therefore, the best model ought to be selected to accurate fall detection. Also, the developed system needs evaluation on its performance in a real world environment to validate its use in a home-based care setting. This chapter highlights the different evaluation metrics used to select the best model suited for a multi-person fall detection event. Additionally, it notes the findings in the failure of other models in the fall detection task. Furthermore, it describes the performance of the fall detection system in a real-time environment providing insights on the actual implementation of the system.

6.2 Model Evaluation

Model evaluation entails the process of assessing the performance of a specified model against a specified metric. Essentially, this enables researchers to distinguish between the different specified model performances. The model configurations developed were evaluated on two distinct metrics, that is, the loss function and the mean Average Precision. Notably, each of the listed metrics define different aspects of the performance of the models as discussed below:

6.2.1 Loss Function

The loss function serves as the indicator of overfit of the model on the provided dataset. The figure below depicts the training loss function against the number of epochs used in the study for the one class configuration model.

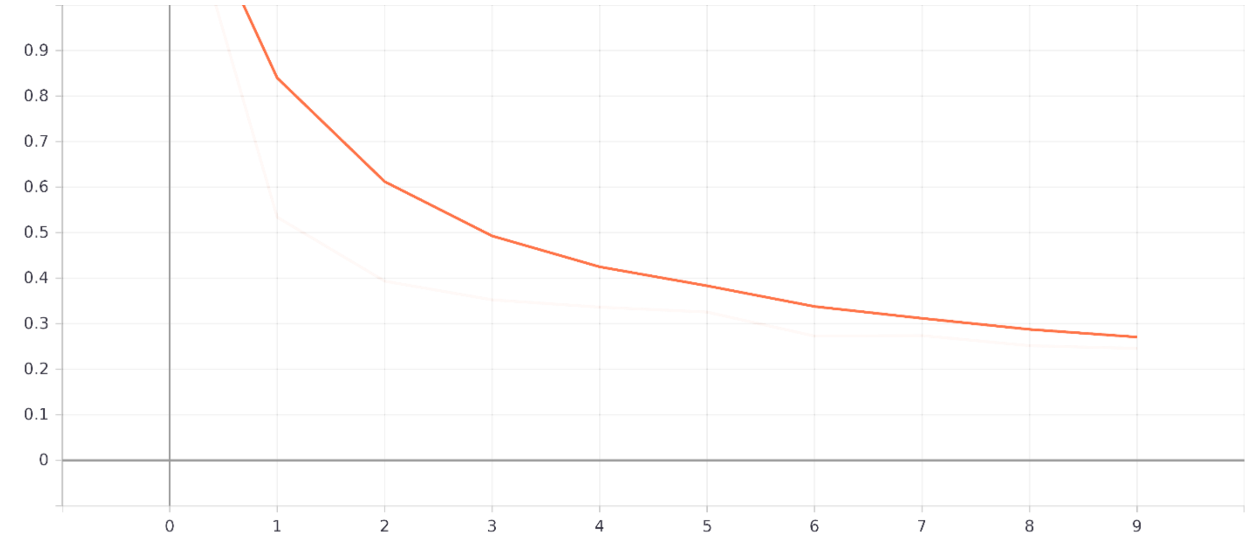

Figure 6.1: One Class Configuration Training Loss

Additionally, to check for overfitting in the model, a comparison of the training loss value with the validation loss function was implemented to aid in the evaluation of the model. The figure below shows the validation loss function against the number of epochs used for the study.

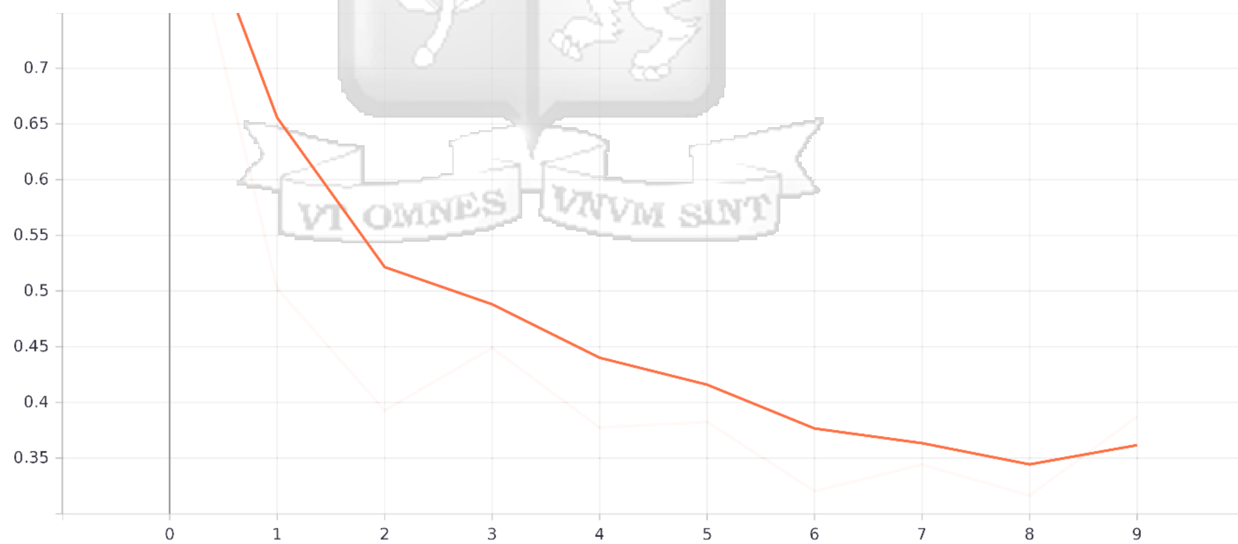

Figure 6.2: One Class Configuration Validation Loss

Notably, a comparison of the two figures depicts that the generated model did not suffer from overfitting with the sample dataset. The final values of the validation and training loss show a 0.1 difference between the training and validation loss.

Also, the two class configuration model was evaluated with the training and validation loss values compared to check for overfitting or under fitting. The figures below depict the training loss against the number of epochs for the model:

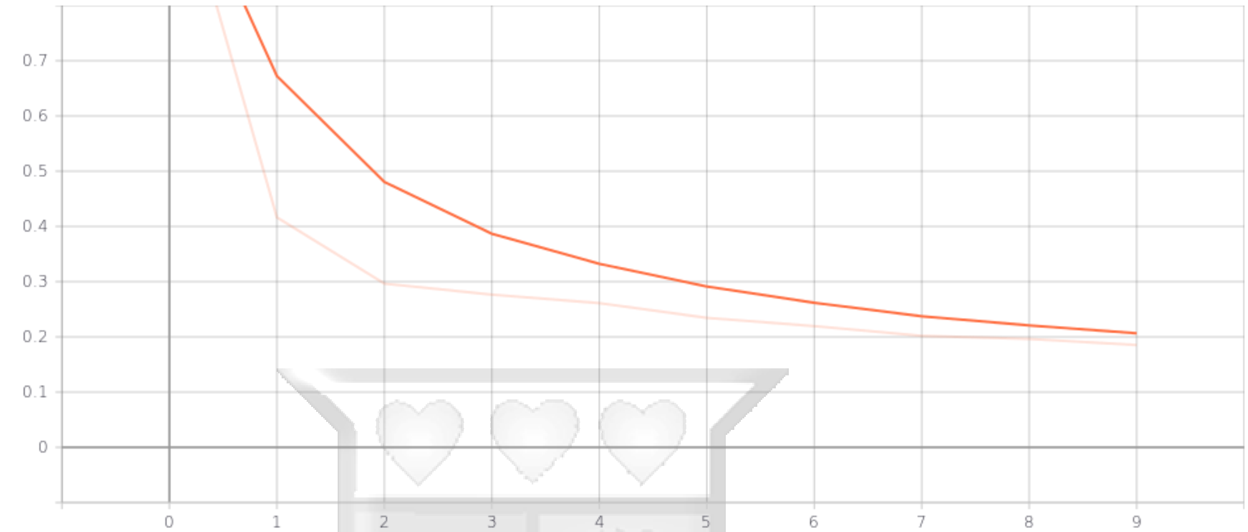

Figure 6.3: Two Class Configuration Training Loss

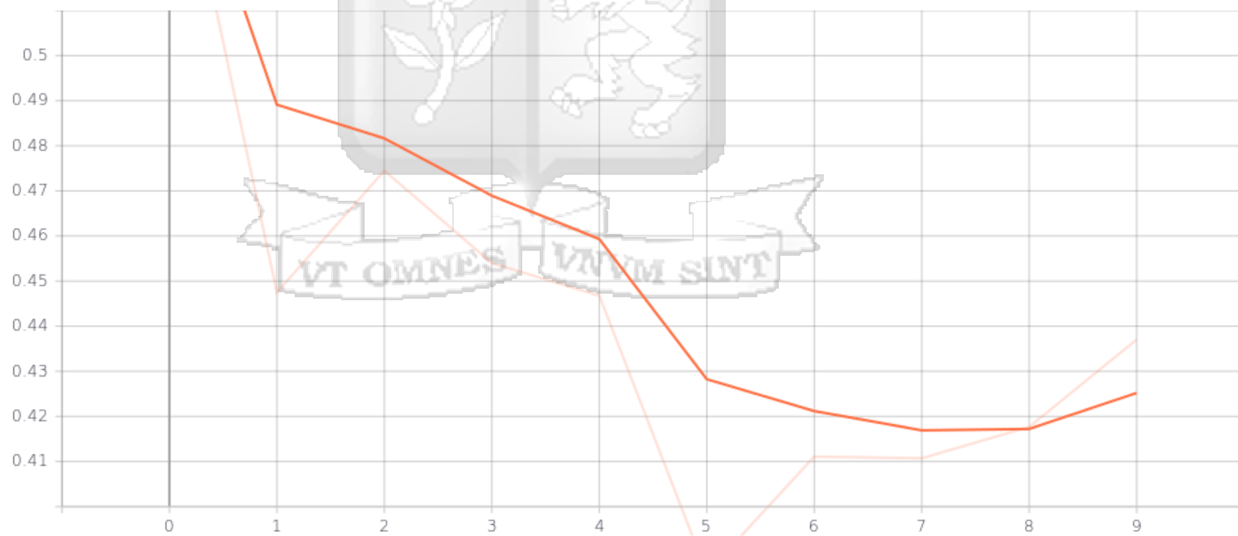

Figure 6.4: Two Class Configuration Validation Loss

As noted above, at the last epoch, the difference between the validation loss and the training loss increased to approximately 0.2. Though the training loss in the model was lower than the value in the one class configuration, the difference between the training and validation loss increased marginally thereby exposing the model to overfitting.

6.2.2 Average Precision

The Average Precision metric serves as the accepted standard of object detection models and will be adopted for evaluating the different model configurations. The Average Precision is calculated as follows:

$$\mathrm{AP} = \sum_{r \in \{0.0, \ldots 1.0\}} AP_r$$

Equation 6.1: Average Precision

where the AP_r represents the average precision of each of recall value multiplied by the precision value. Notably, the precision values can be calculated as follows:

$$Precision = \frac{\mathrm{TP}}{TP + FP}$$

Equation 6.2: Precision

$$Recall = \frac{\mathrm{TP}}{TP + FN}$$

Equation 6.3: Recall

where TP represents True Positive, FP represents False Positive, and FN represents False Negative. The precision and recall values vary differently when subjected to different Intersection over Union Thresholds. Notably, as discussed earlier, a higher threshold value evaluated model may negatively affect the precision value of the model. Essentially, the ideal model ought to have a high IoU threshold value coupled with a high precision value. Therefore, the generated models from the two configurations were tested with different threshold values. The two model configurations were each tested with a sample of the 37 images used as the validation dataset to compute the Mean Average Precision for the generated models. The table below shows the performance of the model configurations using a threshold value of 0.5.

Table 6.1: Model Configuration Threshold – 0.5

Model Configuration	**Mean Average Precision**
One Class Configuration	1.0
Two Class Configuration	0.68

As depicted above, the one class configuration performs better with a mean average precision above normal performance. Essentially, this is due to a low IoU threshold value thereby increasing the precision in each detection depicting better performance. As noted utilizing a low threshold value leads to an inference of high performance which does not represent the correct performance of each of the models. Therefore, to attain a comprehensive performance of the models, the Average Precision was calculated within a range of IoU value between 0.5-0.95. As such this represented a better view of the performance of the model. The table below shows the mean average precision for each of the models with the updated IoU values:

Table 6.2: Model Configuration Analysis

Model Configuration	**Mean Average Precision**
One Class Configuration	0.8
Two Class Configuration	0.25

The one class configuration model performs better than the two class configuration as depicted above. Therefore, the one class configuration model serves as the chosen model for the fall classification event.

6.2.3 Real-Time Evaluation

Essentially, for accurate and responsive fall detection, the system ought to provide real-time analysis of the captured images. As such, the system was evaluated on its capability to provide real time analysis of falls. This was achieved with the use of sample videos with typical fall and non-fall scenarios that provided insights on the performance of the system in a real-time environment. The table below shows the performance of the system on videos with particular durations:

Table 6.3: Real Time Evaluation 1 Second

Video Length (Seconds)	Time Taken(Seconds)	Generation Time
22	350	1second
14	267	1second
75	1177	1second
111	**1794**	
Average Time Taken to Analyze Each Video	**16 Seconds**	

In the table above, the system generates an image after every one second of the video and analyses each of the images for fall detection. Therefore, if an image is generated after each second for analysis, the system takes 16 seconds to detect and send a notification message to the users. Notably, this is entirely slow for a real-time fall detection system since it will lead to a delay in subsequent fall incidences. As a result of the challenge, the system was reconfigured to generate images for analysis after 5 seconds rather than each second. Notably, this would affect the time taken to detect a fall in the images. The table below shows the achieved results:

Table 6.4: Real Time Evaluation – 5 seconds

Video Length(Seconds)	Time Taken (Seconds)	Generation Time
22	94	5 seconds
14	67	5 seconds
75	263	5 seconds

111	**424**	
Average Time Taken To Analyze Each Video	**3.8 Seconds**	

The increase in seconds to generate the fall incidence leads to significant improvement in time taken to analyze the videos, however, this has an overall effect on fall detection due to the gap of 5 seconds between each analysis. Notably, this is acceptable since there is less significant changes in frames with a difference of 1 second. Therefore, the 5 second gap is adequate for proper analysis while reducing the average time to detect a fall event.

6.3 Validity of the proposed solution

Fall detection research has been undertaken for a period of time with the research mostly focused on detecting a fall in the scenario of one person appearing on the image. This has led to high accuracy levels achieved with the analysis of one object in the image, however, the systems do not cater for a real life scenario where in one image multiple persons appear in an image. The proposed system caters for multiple people appearing on an image, therefore, representative of real-life scenarios. Notably, the system provides notification of a fall incident within less than four minutes thereby within the timeline outlined by Pons et al. (2005) thus increasing a person's survival chances in the case of an emergency. Additionally, the system provides notification services with the provision of additional information regarding a fall incident. This ensures the creation of a robust system that enables precise notification and easy update of user information. The autonomous system provides for easier monitoring and alert notification without user intervention thereby reducing the required manpower. Lastly, the system does not invade the personal space of an elderly thereby providing the required comfort in executing their daily activities.

Chapter Seven: Conclusion and Recommendations

7.1 Conclusions

Globally, the life expectancy levels continue to increase in low and middle income countries due to proper healthcare and healthy living. Notably, this results to an increase in the number of elderly people under constant monitoring by a caregiver. Essentially, to ensure comprehensive and proper monitoring of the elderly patients, the use of modern technology is imperative. Over the years, the use of modern technology has resulted in impressive results that improve the quality of life in the community. The design and implementation of the proposed system aims at solving the real-life scenario of fall detection in a multi-object or multi-person environment.

Essentially, guided by the conceptual framework designed in chapter two, the system was designed, implemented and tested, highlighting crucial findings instrumental in the evaluation of the proposed system. This chapter analyses findings undertaken in the course of the study in relation to the expected outcomes set by the objectives in chapter one, provides recommendations on improvements of the system and guiding on future research.

The first objective of the study aimed at highlighting the different characteristics of a human being falling. Essentially, in the literature review different approaches to fall detection depicted the perceived attributes of a human fall. Notably, techniques such as inactivity detection, head shape analysis outlined the different characteristics that define the nature of a human being falling. The information served as the basis for the conducted research.

Fall incidents occur randomly as human beings undertake their daily activities. Therefore, the researcher investigated the impact of the fall incident on the life of an elderly patient. Essentially, relying on gathered literature, falls lead to a negative impact on the lives of elderly patients due to reduced quality of life, death and increased healthcare costs. The research aims at improving response time in case of a fall incident thereby increasing survival chances of an elderly individual or improving the quality of life of the person in a home-based care environment.

The third objective of the study entailed reviewing existing algorithms used in the fall detection techniques. Vision-based devices rely on different techniques to detect a fall: inactivity detection, 3D head shape analysis and body shape analysis. The techniques leverage on hand-engineered feature extraction process and focus solely on one person in an image. Therefore, leveraging on deep learning techniques, automatic feature extraction coupled with object detection enables multi-person fall detection.

The model was developed with a review of different configurations that represent real life scenarios. The configurations separated according to the number of classes created on each of the simulated fall images. The simulated falls images served as input to each of the models thereby providing input for the generation of the model. The models were then evaluated on mean average precision to identify the best performing model. With the best performing model, the system was developed to incorporate fall notification events.

The final objective of the study was to test the performance of the generated model in fall scenarios. The developed system was tested against validation data to check on the performance of the model in identifying fall scenarios in a specific image with multiple people. The final generated model obtained a mean average precision of 0.8 against the validation dataset. Further, the system was tested on its suitability to operate in a real-time environment.

7.2 Recommendations

The developed model accuracy value of 0.8 may provide false positives in a real-time environment due to the dynamic nature of a real world environment. Also, the model requires high end computing resources to operate efficiently in real-time which may not be available in a typical home-base care. This is as a result of incorporating transfer learning in the development of the model. The researcher proposes several recommendations based on the findings made:

i. Train the model with a more robust dataset with many images of falls in a multi-person scenario to increase the mean average precision of the model. This will ensure that the model can easily classify the fall and non-fall scenarios of each of the persons in an image.

ii. Train the model to only learn specific features of a person thereby reducing the computing resources required in running the model. This will improve the efficiency of the model in utilizing resources and improving real time fall detection.

iii. Short Message Services enable fast notification to the users, however, in typical real world environment, the services may act as a distraction to the recipient. Therefore, an alert to an emergency medical service would easily reduce the response time.

7.3 Suggestions and Future Research

Deep learning offers numerous unexplored dimensions that can aid in fall detection. The research implemented a manual approach to image annotation for the images representing the different classes. The researcher believes an automated process can be achieved that generates proposals for the classes present in an image for confirmation and validation. This will aid in fastening the training process of any vision-based approach with different classes in an image. Furthermore, with the use of specific model pre-trained on the identification of only specific classes such as person then a fall classification layer added will aid in the application of the system faster as it reduces the processing time undertaken by the model to ensure the processing of each frame generated to an image. Essentially, this improves the performance of the model in real-time execution increasing the accuracy due to frame by frame fall classification and the time taken to analyze each of the generated images. Lastly, since the system may raise false positives in cases where a patient or patients may be lying down, the researcher suggests a check mechanism where a patient can interface with the system to confirm a fall incident.

References

Abdulla, W. (2017). *Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow*. [online] Github. Available at: https://github.com/matterport/Mask_RCNN

Abreu, C., Mendes, A., Monteiro, J., & Santos, F. (2012). Falls in hospital settings: a longitudinal study, *20:597*–603. Doi: 10.1590/S0104-11692012000300023

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).

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

Blackwell, T. H., & Kaufman, J. S. (2002). Response time effectiveness: comparison of response time and survival in an urban emergency medical services system. *Academic Emergency Medicine*, *9*(4), 288-295.

Boaz, A., & Ashby, D. (2003). *Fit for purpose?: assessing research quality for evidence based policy and practice* (pp. 1-55). London: ESRC UK Centre for Evidence Based Policy and Practice.

Centers for Disease Control and Prevention. (2015). *Cost of falls among older adults*. Available from: http://www.webcitation.org/6hanayMoC.

Centers for Disease Control and Prevention. (2016). *Important facts about falls*. Retrieved form http://www.webcitation.org/6gREJTEgu.

Centers for Disease Control and Prevention. (2016). *Falls are leading cause of injury and death in older Americans*. Retrieved form https://www.cdc.gov/media/releases/2016/p0922-older-adult-falls.html.

Creswell, J. W., & Clark, V. L. P. (2007). Designing and conducting mixed methods research.

Dai, X. (2013). *Vision-based 3d human motion analysis for fall detection and bed-exiting* (Doctoral dissertation, University of Denver).

Ekelund, U., Yngve, A., Brage, S., Westerterp, K., & Sjöström, M. (2004). Body movement and physical activity energy expenditure in children and adolescents:

how to adjust for differences in body size and age. *The American journal of clinical nutrition*, *79*(5), 851-856.

Fleming, J., & Brayne, C. (2008). Inability to get up after falling, subsequent time on floor, and summoning help: prospective cohort study in people over 90. *Bmj*, *337*, a2227.

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440-1448).

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580-587).

Habib, M. A., Mohktar, M. S., Kamaruzzaman, S. B., Lim, K. S., Pin, T. M., & Ibrahim, F. (2014). Smartphone-based solutions for fall detection and prevention: challenges and open issues. *Sensors*, *14*(4), 7181-7208.

Hartholt, K. A., Polinder, S., Van der Cammen, T. J., Panneman, M. J., Van der Velde, N., Van Lieshout, E. M., & Van Beeck, E. F. (2012). Costs of falls in an ageing population: a nationwide study from the Netherlands (2007–2009). *Injury*, *43*(7), 1199-1203.

He, J., Bai, S., & Wang, X. (2017). An unobtrusive fall detection and alerting system based on Kalman filter and Bayes network classifier. *Sensors*, *17*(6), 1393.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961-2969).

Heinbüchner, B., Hautzinger, M., Becker, C., & Pfeiffer, K. (2010). Satisfaction and use of personal emergency response systems. *Zeitschrift für Gerontologie und Geriatrie*, *43*(4), 219-223.

Igual, R., Medrano, C., & Plaza, I. (2013). Challenges, issues and trends in fall detection systems. *Biomedical engineering online*, *12*(1), 66.

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

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).

Khan, S. S., & Hoey, J. (2017). Review of fall detection techniques: A data availability perspective. *Medical engineering and physics*, *39*, 12-22.

Khow, K. S., & Visvanathan, R. (2017). Falls in the aging population. *Clinics in geriatric medicine*, *33*(3), 357-368.

Li, Y., Ho, K. C., & Popescu, M. (2012). A microphone array system for automatic fall detection. *IEEE Transactions on Biomedical Engineering*, *59*(5), 1291-1301.

Lohr, K. N. (2004). Rating the strength of scientific evidence: relevance for quality improvement programs. *International Journal for Quality in Health Care*, *16*(1), 9-18.

Mastorakis, G., & Makris, D. (2014). Fall detection system using Kinect's infrared sensor. *Journal of Real-Time Image Processing*, *9*(4), 635-646.

Mao, A., Ma, X., He, Y., & Luo, J. (2017). Highly portable, sensor-based system for human fall monitoring. *Sensors*, *17*(9), 2096.

Nari, M. I., Suprapto, S. S., Kusumah, I. H., & Adiprawita, W. (2016). A simple design of wearable device for fall detection with accelerometer and gyroscope. *Electronics and Smart Devices (ISESD), International Symposium on* (pp. 88-91). IEEE.

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

Ngu, A., Wu, Y., Zare, H., Polican, A., Yarbrough, B., & Yao, L. (2017). Fall detection using smartwatch sensor data with accessor architecture. *International Conference on Smart Health* (pp. 81-93). Springer, Cham.

Núñez-Marcos, A., Azkune, G., & Arganda-Carreras, I. (2017). Vision-based fall detection with convolutional neural networks. *Wireless Communications and Mobile Computing*, *2017*.

Pons, P. T., Haukoos, J. S., Bludworth, W., Cribley, T., Pons, K. A., & Markovchick, V. J. (2005). Paramedic response time: does it affect patient survival? *Academic Emergency Medicine*, *12*(7), 594-600.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (pp. 91-99).

Rougier, C., & Meunier, J. (2006). 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, C., Meunier, J., St-Arnaud, A., & Rousseau, J. (2007). 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.

Towne, L., & Shavelson, R. J. (2002). Scientific research in education. *Washington, DC: National Research Council*.

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

Tamura, T., Yoshimura, T., Sekine, M., Uchida, M., & Tanaka, O. (2009). A wearable airbag to prevent fall injuries. *IEEE Transactions on Information Technology in Biomedicine*, *13*(6), 910-914.

Tabar, A. M., Keshavarz, A., & Aghajan, H. (2006). Smart home care network using sensor fusion and distributed vision-based reasoning. *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks* (pp. 145-154). ACM.

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.

Terroso, M., Rosa, N., Marques, A. T., & Simoes, R. (2014). Physical consequences of falls in the elderly: a literature review from 1995 to 2010. *European Review of Aging and Physical Activity*, *11*(1), 51.

Thuc, H. L. U., & Van Tuan, P. (2014). An Effective Video Based System for Human Fall Detection. *Int. J. Advanced Research in Computer Engineering & Technology*, *3*(8), 2820-2826.

Wang, C., Chiang, C., Lin, P., Chou, Y., Kuo, I., Huang, C., & Chan, C. (2008). Development of a fall detecting system for the elderly residents. *Bioinformatics and Biomedical Engineering, 2008. ICBBE 2008. The 2nd International Conference on* (pp. 1359-1362). IEEE.

World Health Organization (WHO) (2018). Falls. Retrieved from http://www.who.int/mediacentre/factsheets/fs344/en/

Wibisono, W., Arifin, D. N., Pratomo, B. A., Ahmad, T., & Ijtihadie, R. M. (2013). Falls detection and notification system using tri-axial accelerometer and gyroscope sensors of a smartphone. *Technologies and Applications of Artificial Intelligence (TAAI), 2013 Conference on* (pp. 382-385). IEEE.

Wu, F., Zhao, H., Zhao, Y., & Zhong, H. (2015). Development of a wearable-sensor-based fall detection system. *International journal of telemedicine and applications*, *2015*, 2.

Yoshida–Intern, S. (2007). A global report on falls prevention epidemiology of falls. *WHO. Geneva*.

Yu, X. (2008). Approaches and principles of fall detection for elderly and patient. *e-health Networking, Applications and Services, 2008. HealthCom 2008. 10th International Conference on* (pp. 42-47). IEEE.

Zhang, M., & Sawchuk, A. A. (2011). Context-aware fall detection using a Bayesian network. *Proceedings of the 5th ACM International Workshop on Context-Awareness for Self-Managing Systems* (pp. 10-16). ACM.

Zigel, Y., Litvak, D., & Gannot, I. (2009). A method for automatic fall detection of elderly people using floor vibrations and sound—Proof of concept on human mimicking doll falls. *IEEE Transactions on Biomedical Engineering*, *56*(12), 2858-2867.

Appendix

Appendix A: Use Case Diagram

Table 8.1: Caretaker Registration

Use Case : Caretaker Registration	
Primary Actors : Administrator	
Precondition : Administrator Panel Activated	
Post-condition : Administrator Panel Deactivated	
Main Success Scenario	
Actor Responsibility	**System Responsibility**
1. Administrator enters the individual details of the caretaker.	
	2. The system captures the entered information.
	3. The system validates if all the required data has been correctly entered.
	4. The system saves the information in a database.
5. The administrator exits the administrator panel.	
Alternative 1	
Step 3 fails.	

	1. The system notifies the administrator of the failure of the step.
2. The administrator enters required information correctly.	
	3. The system undertakes step 3 and 4 consecutively.
4. The administrator undertakes step 5.	

Table 8.2: Update Model

Use Case : Update Model	
Primary Actors : Developer	
Precondition: The model should have existing weights.	
Post-condition: The model has updated weights.	
Main Success Scenario	
Actor Responsibility	**System Responsibility**
1. The developer loads the system into memory.	
2. The developer loads new series of images to the system.	
3. The developer initializes the system with the new series of images.	

	4. The system updates its features from the images.
	5. The features enable predictions while new weights are updated in the model.
	6. The system stores the new weights and saves the generated model.
7. The developer exits the system.	
Alternative 1	
Step 5 fails	
1. The developer initializes with new random weights	
	2. The system undertakes step 5 and 6.
3. The developer undertakes step 7.	

Table 8.3: Camera Mapping

Use Case : Camera Mapping	
Primary Actor: Administrator	
Precondition : Administrator Panel Activated	
Post-condition: Camera Panel Deactivated.	
Main Success Scenario	
Actor Responsibility	**System Responsibility**

1. The administrator enters the details of a camera to the system.	
	2. The system captures the details of the camera and saves.
3. The developer exits the camera panel.	

Appendix B: Code Snippets

```
from __future__ import print_function
import mysql.connector
import cv2
import os
import sys
import random
import math
import re
import time
import h5py
import africastalking
import numpy as np
import tensorflow as tf
import matplotlib
import matplotlib.pyplot as plt
import matplotlib.patches as patches
from skimage import io

ROOT_DIR = os.path.abspath("../")
print(ROOT_DIR)

sys.path.append(ROOT_DIR)
from mrcnn import utils
from mrcnn import visualize
from mrcnn.visualize import display_images
import mrcnn.model as modellib
from mrcnn.model import log

from ProjectTwo import code
```

Figure 8.1: Import Functions

```
#ModelConfigurations
get_ipython().run_line_magic('matplotlib', 'inline')
MODEL_DIR = os.path.join(ROOT_DIR, "ProjectTwo")
WEIGHTS_PATH = "/home/andmerc/Thesis/Project/ProjectTwo/mask_rcnn_fall_0010.h5"
config = code.BasicConfig()
DATASET_DIR = "/home/andmerc/Thesis/Data/val"

#DatabaseSettings
hostname='localhost'
username='andie'
password='Guns123'
database='Thesis'
def Querie(conn,cam):
    cur = conn.cursor()
    cur.execute("SELECT Room_ID, Room_Desc FROM Room WHERE Camera_ID=%s",cam)
    record = cur.fetchall()
    RoomID=record[0][0]
    RoomDesc=record[0][1]
    return RoomID, RoomDesc
def QuerieTwo(conn):
    cur = conn.cursor()
    cur.execute("SELECT Phone_Number FROM Caretaker")
    record = cur.fetchall()
    recipient = []
    for i in range (len(record)):
        recipient.append(record[i][0])
    return recipient
def conn(hostname,username,password,database):
    myConnect = mysql.connector.connect(host=hostname,user=username,passwd=password,db=databa
    return myConnect
cam=1 #changed to depict the
connection = conn(hostname,username,password,database)
roomid, roomdesc = Querie(connection,(cam, ))
recipient=QuerieTwo(connection)
connection.close()
```

Figure 8.2: Database and Model Settings

```
class InferenceConfig(config.__class__):
    GPU_COUNT = 1
    IMAGES_PER_GPU = 1
config = InferenceConfig()
#config.display()

DEVICE = "/cpu:0"

TEST_MODE = "inference"

with tf.device(DEVICE):
    model = modellib.MaskRCNN(mode="inference", model_dir=MODEL_DIR,
                              config=config)
    model.load_weights(WEIGHTS_PATH, by_name=True)

pathIn = '/home/andmerc/Thesis/Project/ProjectTwo/falling.mp4'
pathOut= '/home/andmerc/Thesis/Project/ProjectTwo/Rimages'

def extractImages(pathIn, pathOut):
    count = 0
    vidcap = cv2.VideoCapture(pathIn)
    success,image = vidcap.read()
    success = True
    while success:
      vidcap.set(cv2.CAP_PROP_POS_MSEC,(count*1000))
      success,image = vidcap.read()
      print ('Read a new frame: ', success)
      cv2.imwrite(pathOut + "/" + "frame%d.jpg" % count, image)
      count = count + 1
      if count==14:
            success=False
    vidcap.release()
```

Figure 8.3: Image Extraction

```
extractImages(pathIn,pathOut)
cv2.destroyAllWindows()
count = 0
for x in range(14):
    image = io.imread(pathOut + "/" + "frame%d.jpg" % x)
    results = model.detect([image], verbose=1)
    #print(results)
    r = results[0]
    x = r['class_ids']
    print(type(x))
    if (x.size == 0):
        print('No Fall Detected')
        print(count)
        count = count + 1
    else:
        print('Fall Detected')
        print(count)
        count = count + 1
        message = "Fall Detected in rooom " + str(roomid) + ". The " + str(roomdesc)
        #print(recipient)
        msg = SMS().send(message,recipient)
        print(msg)
    del x
```

Figure 8.4: Send SMS

Appendix C: Ethical Approval

Strathmore
UNIVERSITY

23 rd. April 2019

Andrew Kinyua Waruguru,
P.O BOX 152,
Meru.
awkinyua@strathmore.edu

Dear Andrew,

REF **Protocol ID:** SU-IERC0361/19 **Student Number**: 102708

A VISION-BASED APPROACH TO FALL DETECTION FOR ELDERLY PATIENTS RECEIVING HOME-BASED CARE

We acknowledge receipt of your application documents to the Strathmore University Institutional Ethics Review Committee (SU-IERC) which includes:

1. Study Protocol submitted 18th April 2019
2. Cover letter listing all submitted documents 18th April 2019
3. Proposal declaration page signed by supervisors 18th April 2019

The committee has reviewed your application, and your study *"A vision-based approach to fall detection for elderly patients receiving home-based care"* has been granted approval.

This approval is valid for one year beginning ***23rd April* 2019** until ***23rd April 2020***

In case the study extends beyond one year, you are required to seek an extension of the Ethics approval prior to its expiry. You are required to submit any proposed changes to this proposal to SU-IERC for review and approval prior to implementation of any change.

SU-IERC should be notified when your study is complete.

Thank you

Sincerely,

STRATHMORE UNIVERSITY INSTITUTIONAL ETHICS REVIEW COMMITTEE (SU-IERC)
23 APR 2019
TEL: +254 (0)703 034 000
P. O. Box 59857 - 00200
NAIROBI - KENYA

Prof. Florence Oloo
Secretary
Strathmore University Institutional Ethics Review Committee

Ole Sangale Rd, Madaraka Estate. PO Box 59857-00200, Nairobi, Kenya. Tel +254 (0)703 034000
Email admissions@strathmore.edu www.strathmore.edu

Appendix D: Plagiarism Report