Automatic Fall Risk Detection based on Imbalanced Data

by

Yen-Hung (Kevin) Liu

A thesis submitted to the
School of Graduate and Postdoctoral Studies in partial
fulfillment of the requirements for the degree of

Master of Science in Computer Science

In

The Faculty of Business and Information Technology

University of Ontario Institute of Technology (Ontario Tech University)

Oshawa, Ontario, Canada

August 2021

THESIS EXAMINATION INFORMATION
Submitted by: Kevin Liu

Master of Science in **Computer Science**

Thesis title: Automatic Fall Risk Detection based on Imbalanced Data

An oral defense of this thesis took place on July 28, 2021 in front of the following examining committee:

Examining Committee:

Chair of Examining Committee	Alvaro Quevedo
Research Supervisor	Patrick Hung
Research Co-supervisor	Farkhund Iqbal
Examining Committee Member	Miguel Vargas Martin
Examining Committee Member Thesis Examiner	Benjamin Fung McGill University'

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

ABSTRACT

In recent years, the declining birthrate and ageing population have gradually brought countries into an ageing society. In regards to the accidents that occur amongst the elderly, falls are an important problem that quickly causes indirect physical loss. In this thesis, we propose a pose estimation-based fall detection algorithm to detect fall risks. Since fall data is rare in real-world situations, we train and evaluate our approach in a highly imbalanced data setting. We assess not only different imbalanced data handling methods, but also different machine learning algorithms. After oversampling on our training data, the K-Nearest Neighbors (KNN) algorithm achieves the best performance. This experiment provides evidence that our approach is more interpretable, with key features from skeleton information, and workable in multi-people scenarios.

Keywords: Fall Detection; Pose Estimation; Machine Learning; Data Sampling; Anomaly Detection

AUTHOR'S DECLARATION

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize the University of Ontario Institute of Technology (Ontario Tech University) to lend this thesis to other institutions or individuals, for the purpose of scholarly research. I further authorize University of Ontario Institute of Technology (Ontario Tech University) to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

____ Yen-Hung Liu

STATEMENT OF CONTRIBUTIONS

I hereby certify that I am the sole author of this thesis. I have submitted this thesis to IEEE Access, and the paper is processing now. I have used standard referencing practices to acknowledge ideas, research techniques, or other materials that belong to others. Furthermore, I hereby certify that I am the sole source of the creative works and/or inventive knowledge described in this thesis.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deep and sincere gratitude to my supervisor, Prof. Patrick Hung. This thesis was made possible with your guidance and professional advice. As I reminisce my first moments in Canada as your visiting scholar, I am thankful to have worked alongside you towards becoming your master student. You always encouraged my work and led me through any challenges. I would also like to thank my co-supervisor Prof. Farkhund Iqbal, from Zayed University in the UAE. Besides the thesis, I am grateful for the opportunity you have offered me. This allowed me to work with various exciting projects, which widened my vision and enhanced my experience. I am honored to be your student. Thank you for your mentorship, friendship, and inspiration.

Furthermore, I also want to express my strong gratitude towards all of my friends in Canada. I am thankful for their support and comfort throughout tough obstacles. As an international student, it is my pleasure to meet all of you. You all are the treasure in my unmemorable journey. In particular, I would like to give a special thank you to Prof. Miguel Vargas Martin and Prof. Benjamin Fung from McGill University to be my committee members. Last but not least, I want to acknowledge my loving family. Thank you for raising me and supporting me in every perspective. I appreciate the guidance and care you have deliberately offered without the need for anything in return. Thank you for sacrificing your time at work and giving me a life without concern. I am truly grateful for your unconditional support as I study abroad. I recognize the tough decision you have made, but with your heartfelt support, you allowed me to pull through with it. All of you are my heroes. I hope that I can be as successful as you all are in the following stages of my life. I will strive to work harder every day in order to make you both proud.

CONTENTS

LIST OF TABLES

CHAPTER 4

CHAPTER 6

LIST OF FIGURES

LIST OF EQUATIONS

CHAPTER 5

CHAPTER 6

Chapter 1. Introduction

1.1 Motivation

With the advent of an ageing society, population ageing has become a common issue for many countries in the world. However, the biggest impact of population ageing on society is the rapid increase of demand for medical support and long-term care. Sometimes an accident may cause a huge financial burden on a family. According to the World Health Organization (WHO), falls are the second leading cause of accidental death globally, and 37.3 million falls require medical care each year. Among them, adults over 65 have the most life-threatening falls [34]. However, death is not the main result of immediately falling, as it can occur due to various complications caused by falls. Since the elderly often have a high prevalence of coexisting diseases, such as osteoporosis and organ function degradation, even a slight fall may cause great danger. Individuals that live alone, and are 65 and older, make up 24.6% of the Canadian population [43]. If an accident occurs, it is difficult for the elderly to be found. As a result, older adults are more prone to missing their golden treatment time. The shortage of caregivers promotes the health care system to automate. Developing an automatic fall risk detection system can effectively prevent tragedy and reduce the cost. Therefore, fall detection in the elderly is an emerging research field in recent years.

1.2 What is Fall Risk

Fall Risks are common threats that can affect all individuals, including the elderly or young children. Regardless of the victimized person, it can induce significantly harmful and dangerous results. Unfortunately, there are countless ways to assess factors that cause people to fall, but some may include one's bad eyesight, poor balance, use of medications that cause one to be drowsy, and more. Fall risk is important to understand in order to take proper actions for prevention, as injuries associated with this accident can fracture one's physical health.

The significance of fall risk goes beyond identifying who is more at risk. Even though older adults are more susceptible to this injury, the importance is to assess the prevention methods and other valid reasons that cause fall risks. There are many ways to analyze and conduct assessments, such as the study of human behaviour and the utilization of strong technological advancements to monitor movement and balance. This can depict one's actions, triggering specific causes and reasons that may lead to falls. Many studies show different assessment methods and evidence-based approaches to truly deliver accurate data regarding the causes of fall risks and proper ways of prevention.

Regarding the topic of fall risks, many recognized environmental hazards can be another cause for this mishap. We somehow do not have control over these factors. However, factors can be minimized through the practice of cautious behaviour. The flooring situation, such as rugs, carpets, mopped floors, and overly-packed areas, can lead to the dangerous falls among individuals. This shows that falls can occur, regardless of an individual’s personal issues, due to environmental factors.

Although the elderly are more at risk when it comes to falls, younger individuals, such as children, are also prime victims. Falls can happen just about anywhere, but children become vulnerable to their surroundings and environment, especially at a very young age. Depending on the child's physical surroundings, it can lead them to face serious danger, if they are not guarded or being taught how to take proper steps to be cautious. However, when looking at physical factors, as children still are developing stronger balance, this can factor in risks when it comes to falls. Also, those who are weak, and are diagnosed with a certain illness, are at risk of this injury.

Taking proper preventive measures when it comes to fall risk will allow patients to be safe from harm and away from hospitals. However, it is impossible to fully prevent falls, as many possibilities could factor in some type of risk. This could be an individual’s occupied space and area. It is important to identify an unsafe place by looking at how the place is structured and framed. Perhaps, staying in an area with handle bars and other tools to hold on could be a helpful tactic to prevent falls. It is a significant tool to assess one's physical surroundings, as anyone could be susceptible to this injury.

Fall risks are an important topic to discuss and assess, as it has many sets of factors that can cause it to happen, endangering many people's lives. Methods and other applications are utilized in many different settings to predict a patient's risk of falling. As stated before, elderly people are put at risk for this injury due to the changes in their body frame, physical and mental health, as well as cognitive alterations. The level of risk depends on the individual, their physical environment, as well as their lifestyle. Some individuals enjoy adventure, but living a safe lifestyle is always a better choice, as it can prevent falls and other medical issues.

Fear of falling is an issue that individuals are prone to feel after their fall incident. It can change one’s perspective on their surroundings and may also be another reason for falls to occur again. [27] This is due to the trauma that can affect their mental health. Falls can happen anywhere. The injury of falls can result in broken body parts. However, fatality can be an outcome of severe falls. It is very important to shed light on the issue of falls, as it can be more serious than what others may imagine. It is a risk that many researchers are still assessing in order to identify the proper steps to prevent it from occurring. Although older people are more prone to this type of risk, children and other individuals can also be prime victims. Falls are serious, and this risk can be prevented once people understand their dangers. Fall risk is an important subject to study due to its strong relevance in the lives of people to this day. It is a risk that can be prevented, but individuals need to put the effort in to protect themselves.

1.3 What is Machine Learning

With the advent of Big Data, the amount of data increases exponentially, resulting in the increase of complex issues. However, the utilization of data for the purpose of solving more complicated issues, and discovering important insights, is a widely applied trend in the world. This is the main reason for the increasing popularity of machine learning. Machine learning is one of the fields in Artificial Intelligence, that combines computer science and statistical analysis. Machine learning can solve knowledge problems and help with decision-making by having a strong self-understanding of the structure, and correlation of the data. Regarding machine learning, we can divide it into two categories, supervised learning and unsupervised learning.

In the general machine learning process, we split the dataset into training data and testing data. Training data is for training the machine learning model, and the testing data is for evaluating the model performance. The dataset can also be divided into labelled data or unlabelled data. If the data is labelled, it indicates that we know the answer of the data. In this case, the machine learning model can improve its accuracy by comparing it with the answer. The biggest difference between unsupervised learning and supervised learning is whether the training data are labelled or not.

In supervised learning, we can further divide the model into a regression model and classification model. The predicted result for the regression model are continuous values, and for classification, models are categorical values. Since the data are labelled, the model can evaluate the performance with the answer and improve iteratively. As for unsupervised learning, it tends to discover the hidden patterns and rules in the data. Unsupervised learning is mainly used on clustering problems. Since the datasets are unlabelled, the model focuses more on the data's features and correlation.

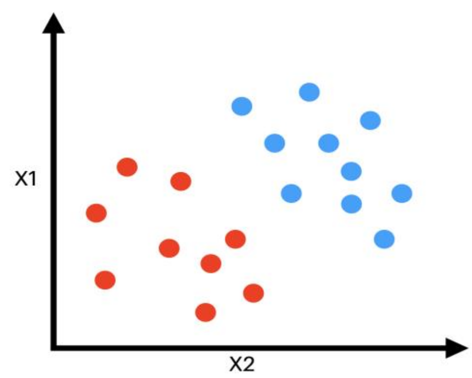


Figure 1.1: Supervised Learning with labelled data

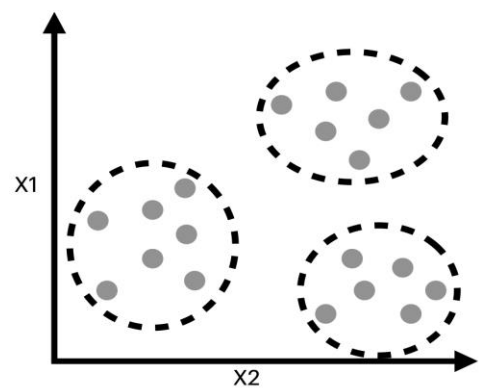


Figure 1.2: Unsupervised Learning with unlabelled data

Machine learning can be utilized in different fields. In business fields, the recommendation system can be completed by tons of user records. With the history of information, the bank is able to identify the possible individual who may cancel the account. In the healthcare field, physical assessment, medication recommendations can be made by machine learning. The wearable sensor collects the physical information from the patient, and the information can be used to assess their physical status. Medication recommendations can assist doctors in making diagnoses with an individual's symptom and medicine information. From assessment to diagnosis, to treatment, machine learning can be utilized everywhere. In industry fields, the status of different machines can be recorded and analyzed. Machine learning can be used to detect any defect that occurs or optimizes the process.

The innovation of machine learning-based applications is continuously developing. Machine learning is changing the world from multiple perspectives. As stated before, machine learning can handle jobs that require specific domain knowledge, which can save on labour and provide more automation. Machine learning can solve more complex problems and customize them to different situations, whether it be in the automation process, compared to traditional computational algorithms.

1.4 Current Issues in Fall Detection Research

In the previous research, fall detection research can be divided into wearable-based methods, ambient-fusion-based methods, and vision-based methods. In wearable-based methods, they make use of accelerometers and gyroscopes to obtain relevant data and make predictions. Although this method can perform in real-time and has no privacy issues, the elderly's views on wearable devices will be uncomfortable, and inconvenient to wear for an extended period of time. As for ambient-function-based methods, the type of method combines multiple sensors to obtain environmental data and perform detection. This method's advantages result in fewer privacy issues and is less intrusive, but its performance is easily affected by external factors. In this case, the false alarm rate is high. As for vision-based methods, the advancement of image processing capabilities and Convolution Neural Networks (CNN) bring computer vision to a new level. However, although CNN can get high accuracy in many computers vision tasks, CNN is like a *Black Box*. The decision-making is hardly interpretable. The advantage of this method is more convenient and non-intrusive, but the disadvantages are privacy issues and interpretability.

From the dataset perspective, obtaining fall data often has privacy and moral restrictions, so most available fall datasets are recorded in experimental environments. For example, Shehroz et al. [10] questioned whether simulated fall data could represent real-life fall events. Since falls are rare events in real life, a data imbalance problem is present.

1.5 Thesis Contributions

In this thesis, we propose a pose estimation-based, fall detection algorithm. We use the pose estimation algorithm, OpenPose, to extract skeleton information and transfer them into interpretable features. Then, we train our model by machine learning methods. We collect four public fall datasets, along with one gait dataset. We divide them into three different classes, Normal, Fall, and Lying. The class distribution is highly imbalanced. The purpose of this study is not only to propose a more interpretable vision-based method, but also to evaluate the fall events in an imbalanced data perspective. Since surveillance cameras are everywhere, our approach is more suitable for the current society.

Our contribution can be summarized as follows:

- We evaluate our approach from an imbalanced data perspective that meets the real-world situation.
- We use skeleton information as features that can have fewer privacy issues.
- Our approach can work in multi-people scenarios.
- By transferring the skeleton information into the interpretable feature, our approach is more interpretable.

1.6 Thesis Organization

This thesis is organized into seven chapters. Chapter 2 details the background of fall events, and more statistics information regarding this problem. Chapter 3 describes the literature review in the correlated area and more detailed information. We compare the pros and cons of the different-based approach. Chapter 4 discusses the dataset collections. We introduce four public fall dataset and one gait dataset. Chapter 5 shows our fall detection architecture. We present the whole structure of our approach, feature preprocessing, and how to handle imbalanced datasets. Chapter 6 shows the experiment and the process in different scenarios. The experiment results will be shown in this section. Chapter 7 discusses machine learning performance, the pose estimation-based approach's pros and cons, the comparison with the previous vision-based work, and further improvement recommendations. Lastly, chapter 8 is a summary of the entire article.

Chapter 2. Background

2.1 Definition of Fall

In this thesis, our definition of a fall event follows the Prevention of Fall Network Europe (ProFANE) [2], described as follows. A fall is an accident that caused the participant to stop on the ground floor or lower [2]. Falling events caused by external force events, such as car accidents, loss of consciousness, and epilepsy, are not in our discussion scope.

2.2 Statistics Data for Fall

In recent years, preventing falls is a significant health indicator for the medical systems of various countries. Falls can occur at any age, but the elderly are the most common victims of this accident, leading to more serious consequences. According to Canada's Public Health Agency statistics, about 20%-30% of older people in Canada fall every year [1], while institutionalized older people are more than 40% likely to do the same. [4] In 2003, 47.2% of every thousand elderly people were injured due to falls. This proportion has increased each year, rising the rate to 57.5% in 2009/2010 [1]. According to gender data, females have a higher chance of falling and a larger hospitalization rate than males. The possible reason is that women have a higher chance of getting osteoporosis. More importantly, these datasets are positively correlated with age growth.

2.3 Common Fall Situations and Fall Types

In a majority of fall events, walking accounted for 60%, while going up and down stairs accounted for 13% [1]. Most of the locations are either at homes or nursing institutions, but the proportion at home is 50% [1]. The direction of the fall is mostly side falls. The severity of a fall to the previous test side will be more severe than a fall in the back. If an individual's walking speed is fast, the direction of the fall will mainly be a forward fall. However, if the person's walking speed is slow, the direction of the fall will mainly be a fall to the side or back. For patients with pelvic fractures, most of the falls are to the side [5][6][7].

2.4 Related Fall Injuries

Among fall-related injuries, sprains or strains accounted for 30%, bruises and contusions accounted for 19%, and fractures accounted for 35% [1]. Although most injured areas are on the shoulders and hands, a few parts can cause serious effects. Among them, head and hip injuries are the most serious. In all injured parts, head injuries accounted for 7%, but most head injuries in children and the elderly are related to falls. In regard to fatality rates caused by falls, head injuries accounted for 46% [1][3]. Hip injuries account for 7%, but 90-95% of hip fractures are caused by falls [1][3]. Among the elderly with hip fractures, 76% of the patients will have limited mobility and reduced daily living ability in the future. Also, 22% of patients will be moved to nursing homes [3].

On the psychological level, the fear of falling will also affect the daily routine of the elderly. Obstacles to basic living abilities can also affect the mental state of the elderly, causing a decline in self-confidence and even depression. Such a vicious circle can also increase the risk of falling again in the future.

2.5 Possible Factors for Fall

The cause of falls are usually multifactorial. The risk factors for falls can be divided into inherent factors and external factors. The inherent factors are related to personal characteristics, such as age and gender, and some chronic diseases or physical problems. The external factors are related to the environment, such as the floor's material or the brightness level of a specified area or place.

Abnormal gait and balance disorders are considered one of the most serious risk factors among the inherent factors [2][4]. Due to ageing and chronic diseases, the strength of the lower limbs of the elderly decreases, and the gait tends to appear slow or stiff. Some medications that cause dizziness and drowsiness can also affect the elderly's gait and balance. In initial fall risk assessment, the history of fall and frequency of fall is needed to decide whether the patient requires a comprehensive assessment or not. However, as long as the elderly have gait or balance problems, a comprehensive assessment is required.

2.6 Machine Learning on Health Care

Recently, machine learning has been identified as an important technical application in the field of healthcare. Since the global medical cost is increasing, machine learning was thought to save future costs [41]. With the advance of wearable devices and medical equipment, tons of data is available for use. Some utilization cases of machine learning in health care are assisted or automated diagnoses. It can also be used to identify patients who are prone to recurring illnesses [41]. Machine learning is used in the assessment of the risk of falling [42]. Multiple inherent and external factors can cause falls to occur. Machine learning can be a great pattern recognization and help the physician find the hidden patterns.

2.7 Conclusions

Falls have a severe impact on the health and quality of life, of the elderly. The most common location is one’s home, regardless of whether we think it is the safest. In the proportion of falling directions, most elderly people fall from the side, and the head and hips are likely to cause fatal injuries. Even if the fall's reasons are multifactorial, gait and balance are more important than other factors. With the advent of an ageing society, it is necessary to effectively prevent falls and seek medical attention, as soon as they occur. Since the machine learning application in health care is increasing, the automatic fall detection system can improve the sensitivity and responsiveness of fall events [40]. In addition, some technologies have been developed to mitigate the consequences of fall events. We describe them in the next section.

Chapter 3. Related Work

Previous fall detection research can be divided into two categories: vision-based methods and sensor-based methods. Vision-based methods can also be divided into using Red, Green, and Blue (RGB) images or depth images. Sensor-based methods can be divided into wearable sensor-based methods and ambient fusion-based methods. In terms of the proportion of research, wearable sensor-based methods have the most significant proportion, followed by vision-based methods. The last is the ambient fusion-based method. We introduce the sensor-based method in the following subsections. Additionally, we will also discuss a more detailed review of the vision-based method.

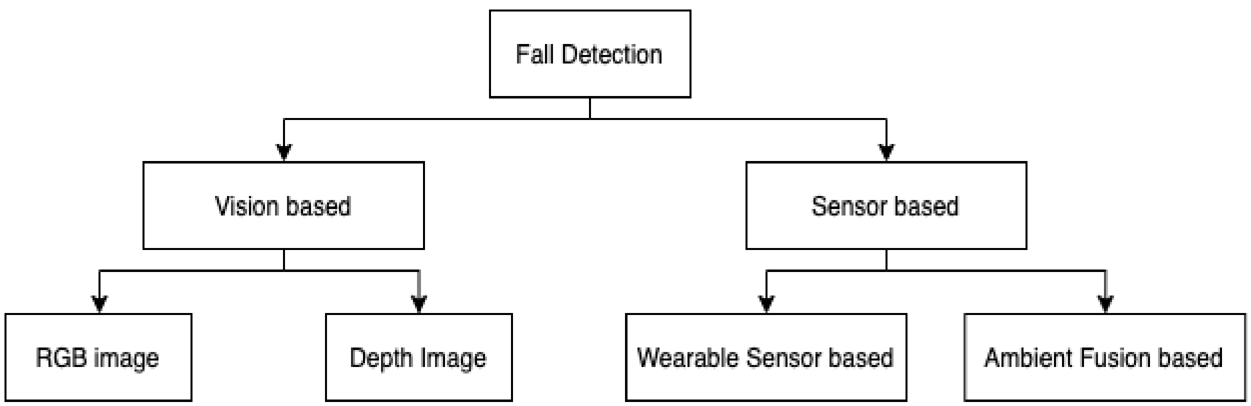


Figure 3.1: Taxonomy of Fall Detection Methods

3.1 Wearable Sensor-based

With the prevalence of wearable devices, an increasing amount of people have started to invest in wearable device research. Wearable devices' fall detection system uses the sensor to detect one's body motion and status. The most commonly used sensor is accelerometer and gyroscope [31].

In the accelerometer method, Perry et al. [8] compared the method without the accelerometer and the method with the accelerometer. The experimental results show that the false alarm rate of the method without an accelerometer is relatively high. However, the position of the accelerometer is also significant. For example, Kangas et al. [9], placed the accelerometer in various body positions. The results showed that the false alarm rate was the lowest when placed on the waist, while the false alarm rate was higher on the head and wrist. However, because their detection methods are mainly identified by setting the threshold, this method usually has a higher false alarm rate. In the gyroscope method, the gyroscope can get the angular velocity information to get the body orientation. This type of method is usually combined with the accelerometer for classification. In the study of Wu et al. [11], combining two kinds of sensors could further improve accuracy. A section of the research primarily focuses on mobile phones because these gadgets are a necessity in daily, human life. In fact, there are more sensors that can be placed on mobile phones [38][39][40]. However, the main disadvantages include battery consumption and insufficient memory. In some cases, only high-end mobile phones are equipped with these sensors.

The advantage of the wearable device method is that the overall development is relatively mature. Their high accuracy can be used indoors and outdoors, and the setup is not complicated. The disadvantage of the wearable device is that power consumption and computing power are limited. Due to the fact that this wearable device needs to be worn for a long period of time, its weakness is that the elderly often forget to wear it. They may also use it in a way that makes them feel uncomfortable. Compared with the other two methods, it is the most intrusive.

3.2 Ambient Fusion-based

The ambient fusion-based method usually requires setting up various sensors around the environment. These sensors include vibration sensor, acoustic sensor, pressure sensor, infrared sensor, doppler sensor, and near electric field. Usually, these sensors are used to cooperate with other sensors.

Vibration sensors and pressure sensors are usually the most common methods. Vibration sensors are generally placed on the floor. For example, Werner et al. [12] believed that the vibration generated by a fall event is different from the Activity of Daily Living (ADL) event. The pressure sensor can be placed in any position, but the distance will affect the pressure's strength. Next, Daher et al. [13] used pressure sensors to form smart tiles, but this method can only detect the fall in which the acceleration is relatively large. However, this is not the case for the slow fall. In the research of acoustic sensors, it is tough to obtain the data. Most of the fall data are obtained by rescue dolls. Then the hardness of a doll is different from that of a human. Moreover, everyone's weight is different in the real world. The method of using the acoustic sensor can only detect hard falls.

The ambient, fusion-based approach's advantage is that it is less intrusive to people, and has less privacy and security issues. However, fall detection can only be detected in a specific environment. In previous research, most of the research solely focused on single-person fall detection, and there was no way to cope with a multi-person environment. Although the ambient fusion-based approach combines more environmental factors, the actual situation often contains other unpredictable factors. Moreover, it is more complicated in installation and setup. The high false alarm rate is a challenging shortcoming.

3.3 Vision-based

With the popularity of surveillance systems and computer vision advancement in recent years, vision-based methods have become a hot research field. We can detect the human body with different computer vision techniques. Traditional computer vision can extract the body contour with background subtraction and track the body movement with Optical flow [55][56]. Deep learning-based methods, Object detection, can recognize the human and surrounded object efficiently [54][55]. Although detecting human objects is not hard with available techniques, identifying the activity such as falls becomes a challenging problem. Since the human body contains different parts which can move freely, some research focuses on the specific body part to design their methods [52][53][57]. Bosch et al. [52] use head, waist, and feet to extract the key features. Hazelhoff et al. [53] use the speed of the head to identify fall events, which has fewer occlusion problems, due to the fact that the head is visible more frequently. Also, the camera can monitor a wide range of areas, and it is contactless. In general acceptance, the vision-based method is the most favourable. Regarding cameras, different types of camera sensors can extract different features from the image. The image data acquisition methods can be divided into the following two categories: RGB image and depth image.

3.3.1 Depth Image

The most common research equipment for depth images is the Microsoft Kinect sensor. Kinect is a low-cost device that uses an infrared projector, combined with an RGB camera to extract depth information. Kinect is used to detect human body movement, and light conditions do not affect its performance. In the study of Volkhardt et al. [14], they installed Kinect on the robot and used different feature extraction and classification methods to determine fall events. Among the classification methods, SVM performed the best. Apichet et al. [15], proposed a new bounding box framework called Directional Bounding Box (DBB), based on a depth camera and Microsoft Kinect. Using Kinect's keypoint and depth information, they rotated the key point to get the appropriate angle to form the DBB. The identification method was easy to cause false positives due to different camera angles using the height and width ratio, to identify in the past. But now, the combined depth information, height-width-depth ratio, and center of gravity are used to identify the fall event. Kinect can also extract the skeleton information. Thi-Lan et al. [59] used skeleton information extracted from Kinect as features. In fact, they made use of the SVMs as a classifier to identify the fall events. Even though the Kinect sensor obtains the most information, there are still some drawbacks. Kinect can work in a dark environment, but is very sensitive to sunlight. This indicates that it is unsuitable for the outdoor environment. The distance detection depth is limited, making it challenging to monitor wide areas [58].

3.3.2 RGB Image

RGB cameras are fairly cheap and easy to set up. RGB cameras have a wider field of view, so most surveillance cameras are RGB cameras. Although the depth information is lacking, the most common vision-based approaches are the RGB cameras. Traditional computer vision methods are usually used for background subtraction, capture body contour, and use tracking techniques in head and shape change. In addition, machine learning methods are used as classifiers. The emergence of Convolutional Neural Networks (CNN) has brought feature extraction to a new level. One of the CNN applications is object detection. One image can include multiple objects, which belong to different classes. The object detection algorithm can use the bounding box to capture the object and identify its class, which is widely used in facial recognition and defect detection. Some well-known object detection algorithms are You Only Look Once (YOLO) and SSD-MobileNet. Both can work in real-time performance. Kun-Lin et al. [16] use YOLO V3 to design their fall detection system. They focused on detecting general fall events and falling events from sitting to standing posture, and events where the body is blocked after a fall. Fall accidents in the elderly, such as sitting down, getting up, and leaving the chair, account for the majority. Since YOLO V3 can detect other objects, they consider the relationship between humans and chairs. In their fall detection system, the first uses YOLO V3 to detect people and chairs. Then, it uses Continuously AdaptiveMeanShift (Camshift), to constantly track the human body and build fall detection algorithms. The system can handle a situation where the chair blocks an individual’s body. Kiran et al.[60] use YOLO as a detection method, as it is based on the height and width ratio of the bounding box to identify fall events. However, humans are a special category in computer vision. Although object detection can capture the human body, the information is not enough to understand human motion. The performance of this approach heavily relies on the camera angles. To overcome this problem, pose estimation can be the solution.

3.3.3 OpenPose

Pose estimation is a computer vision technique used for identifying human postures. Pose estimation can detect a human's body skeleton, which can be used to identify human activity. OpenPose [62] is a well-known pose estimation method used a lot in human action recognition. It can efficiently detect multiple people, and the processing time stays stable when the number of people in the image increases. The whole OpenPose network architecture is shown in Figure 3.1. The whole OpenPose model structure includes two stages. First, they extract the features from the image through convolutional networks (10 layers of VGG-19). VGG-19 is one of the convolutional network architectures. Then, they use the image features as input and send it to the first stage. In the first stages, the convolutional networks predict the part affinity fields. Part affinity is a 2D vector, which associates the different body parts. Part affinity fields can make the model understand the orientation of the limb, and it can help estimate the body part in the second stage. In the second stage, the convolutional networks predict the confidence map with part affinity fields. Confidence maps detect each individual's body part location. With the part affinity fields and confidence maps, OpenPose can efficiently form a body skeleton.

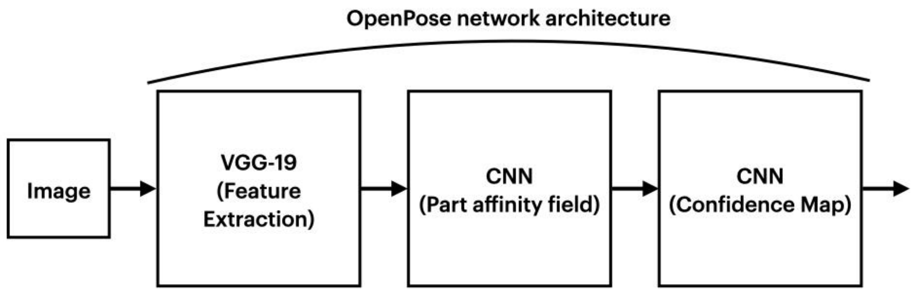


Figure 3.2: OpenPose Network Architecture [74]

Compared to the bounding box, the skeleton is more suitable for detecting fall events. Therefore, some researchers start to use OpenPose as input to boost their research. Chen et al. [17] used OpenPose to extract the human skeleton information based on the unstable center of gravity and symmetry collapse, when a fall event happens. They proposed three key features to identify a fall. These three parameters are the descent speed of the hip joint center, the centerline's angle of the human body to the ground, and the human body's height-width ratio. In their research, we found that the rate of speed descent of the hip joint center significantly influences the prediction of fall events. In addition to general fall events, they also considered whether people could stand up independently after falling. Guangmin et al. [18] used OpenPose plus Single Shot MultiBox Detector (SSD)-MobileNet for their fall detection. The function of SSD-MobileNet is to avoid Openpose's false detection of non-human objects. As for how to identify a fall event, they used Support Vector Data Description (SVDD) classification. The experiments show that their method can effectively reduce the false positive rate in a complex environment [18]. Zhanyuan et al. [19] used OpenPose to perform preprocessing first to obtain pictures with keypoints, as well as the keypoint's coordinates. Then, they put the two into different model identification. In images with keypoints, they used the Visual Geometry Group 16 (VGG-16) for transfer learning and then binary classification to identify fall events. In the keypoint coordinate, Support

Vector Machine (SVM) and Gaussian Kernel are used for identification. Finally, combining the two results effectively improved sensitivity. Most of the fall data is mainly videographed, so it can also be regarded as a time sequence data. Sungil et al. [20] used OpenPose extract skeleton data and then used Long Short-Term Memory (LSTM) to determine fall events. Their approach extracts the coordinates of the shoulders, buttocks, knees, ankles, and the acceleration of these parts as features. In their fall detection, they also divided fall events into "ADL," "Falling," and "Lying." In their experiments, the acceleration of a specific part, and the two states of "Falling" and "Lying," have the highest impact on accuracy.

In the vision-based method, the advantage of using depth images is that more information can be obtained, but people have to stand within a certain distance. For example, Kinect can only detect people within 3 meters. However, the RGB camera detection range is vast, and most current surveillance systems use RGB cameras. The advantage of the vision-based method is that it is more convenient and less intrusive. However, the disadvantage of this method is that it requires more computing resources and it acquires privacy issues. Moreover, light, occlusion problems, and different background conditions have always challenged computer vision.

Regarding three different based approaches, each of them has its own pros and cons. Wearable sensors can detect detailed human status, but are most intrusive. Ambient sensors are easily affected by the environment. Vision-based can capture multiple people, but is easily influenced by occlusions. Despite the data acquisitions being different, fall event classifiers are based on a threshold or machine learning. Since people have different body shapes, setting the threshold is challenging, and the same threshold may not work on different people. [10][36][37]. Therefore, some researchers are focused on machine learning methods, which have better accuracy [31]. At present, most of the fall detection products on the market are mainly sensor-based devices [37]. Depending on the application and use case, we assume vision-based approaches have more potential. Depending on the application and future, we undertake vision-based approaches that have more potential. Vision-based approaches have a wide range of views that can monitor multiple people. In general situations, we identify the fall events through vision. This is an intuitive and more natural element for a detection system.

Therefore, we consider the above drawbacks and design our own approach. First, we use a vision-based approach, which is less intrusive and has more potential in future works. Second, we use skeleton information as features instead of bounding boxes, which understands a more detailed human posture. Third, we use OpenPose instead of Kinect, because we want to apply it in a wider view. Finally, the occlusion problem has less influence on us since even partial body parts are missing. We can still make the prediction through the remaining skeleton.

Chapter 4. Dataset Preparation

Most previous research uses only one dataset to train the model in the dataset collection, performing excellent evaluation results. However, when the input data becomes other fall datasets, the accuracy drops. This indicates that the fall detection model in previous research cannot be applied in general situations. Thus, in our approach, we collected the most commonly used University of Rzeszow (UR) Fall dataset [21], SisFall Dataset [23], Universidad de Malaga (UMA) Fall dataset [22] and Multiple-Camera Fall datasets [24] and divided them into ADL events and fall events. Moreover, since fall is a rare event in real-world situations, we further add one gait dataset, "Aplicaciones de la Visión Artificial" (A.V.A) Multi-View Dataset [25], to make the dataset highly imbalanced. We choose this gait dataset because gait has a high correlation with fall events, and we assume the most common human activity is walking. The major reasons we chose the above datasets are their quality human activity, complete human object outline and most scenes that include only one person.

4.1 UR Fall Detection Dataset

UR Fall Detection Dataset [21] contains 70 videos, including 30 of them being Fall events, and 40 are ADL events. The video resolution is 640 x 480, and the Frame per Second (FPS) is 30. For Fall events, the videos are recorded with 2 Microsoft Kinect cameras, and one is parallel to the floor, while the other is mounted on the ceiling. Fall situations include falls while walking and falling from a chair, and the direction is mostly side falls and forward falls. As for ADL events, the videos are recorded with one camera parallel to the floor. ADL situations include sitting, bending, squatting, grabbing something from the floor, and lying on the bed. In our application, we only use videos that are parallel to the floor.

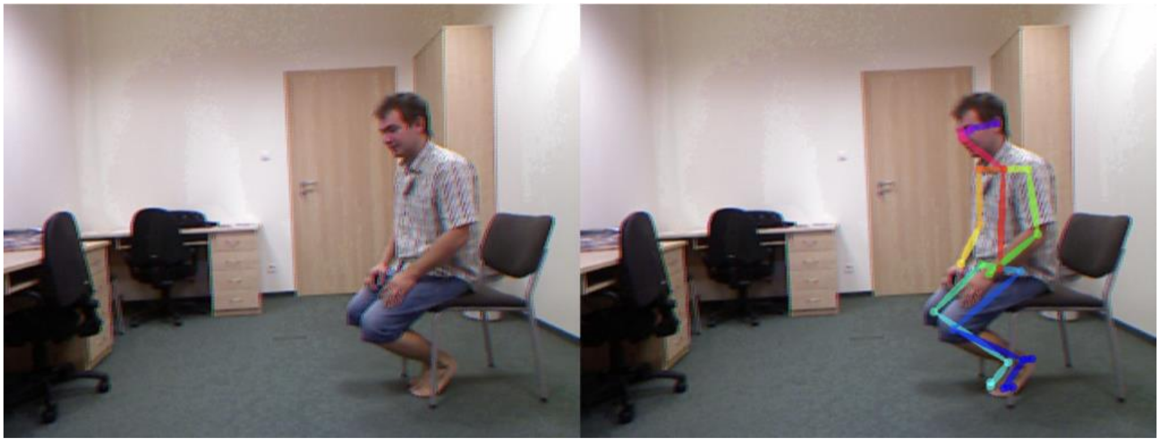


Figure 4.1: Results of UR Fall Detection Dataset [21]

4.2 UMAFall

UMA Fall dataset [22] contains 11 videos, including 3 Fall and 8 ADL events. The video resolution is 854 x 480, and FPS is 30. Fall situations include side fall, forward fall, and backward fall. ADL situations include bending, hopping, jogging, sitting, and walking. Unfortunately, some video angles cannot fully capture the whole body of the tester, so we did not include those videos in our dataset.

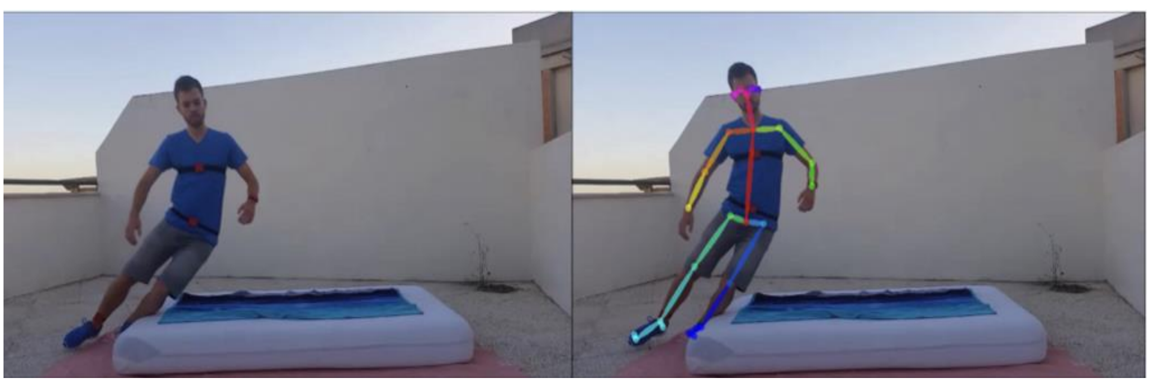


Figure 4.2: Results of UMA Fall Dataset [22]

4.3 SisFall

SisFall dataset [23] contains 34 videos, including 15 Fall events and 19 ADL events. The video resolution is 1920 x 1080, and FPS is 30. It includes a more detailed scenario for fall events, including slip, trip, fainting, etc. The fall direction includes side fall, forward fall, and backward fall. As for ADL events, it includes walking, jogging, walking up and down the stairs, etc. The way they recorded the video is different from other datasets. They moved the camera parallelly and followed the tester instead of setting the camera in a fixed place.

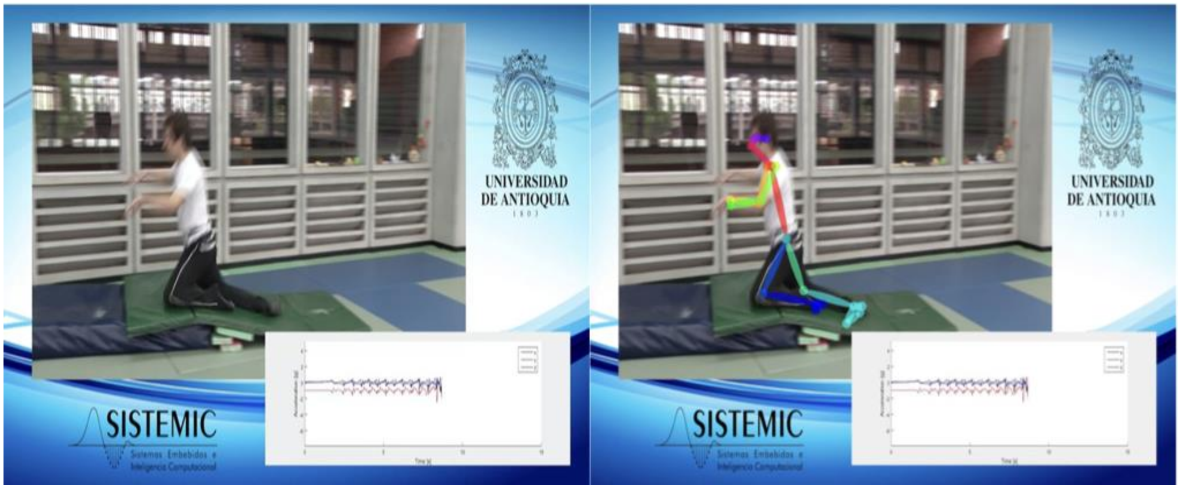


Figure 4.3: Results of SisFall Dataset [23]

4.4 Multiple Camera Fall Dataset

Multiple-Camera fall datasets [24] used 8 cameras to capture different angles. The author did not divide the video into two categories. Instead, the videos start with the tester performing some ADL events, and then the tester falls during the sessions. The ADL events contain walking, housekeeping, and activities that are similar to fall. The fall events have side falls, forward falls, and backward falls. The factors include inappropriately getting up, sitting down, or losing balance. The whole dataset contained 24 different scenarios and was recorded by 8 cameras. Therefore, in total, the number of videos is 192.

Figure 4.4: Results of Multiple Cameras Fall Dataset [24]

4.5 AVAMVG Dataset

The AVA Multi-View Dataset [25] for Gait Recognition (AVAMVG) used 6 cameras to capture different angles. The video resolution is 640x480, and FPS is 30. This dataset only collects the walking event, which is the most common event in our daily life. For the tester, there are 4 females and 16 males participating in 10 different walking activities. Some activities are walking in a straight path, and some are walking in a curved path. Due to the windows' ambient illumination, the video's brightness is different based on the camera's location. The total number of the video is 1200.

Figure 4.5: Results of AVAMVG Dataset [25]

	UR-Fall	UMA-Fall	SisFall	Multiple Camera Fall	AVAMVG
Data type	RGB Video (640x640, 30fps)	RGB Video (854x480, 30fps)	RGB Video (1920x1080, 30fps)	RGB Video (720x480, 30fps)	RGB Video (640x480, 30fps)
Number of Fall and ADL	Fall:30 ADL:40	Fall:3 ADL:8	Fall:15 ADL:19	Video:192	ADL:1200
Camera View Point	1	1	1	8	6
Scenario	Side Fall, Forward Fall	Side Fall, Forward Fall, Backward Fall	Side Fall, Forward Fall, Backward Fall	Side Fall, Forward Fall, Backward Fall	Walking

Table 4.1: All Datasets Description

We have 1507 videos in total. 240 are Fall videos, and 1267 are ADL videos. The number of videos is imbalanced, which meets the real-world situation. However, since the length of each video is not the same, we use the number of frames to represent the ratio of the data in the following sections.

Chapter 5. Fall Detection System Architecture

5.1 Architecture

In our approach, our image data comes from surveillance cameras. First, we use OpenPose to extract the skeleton information from the images. Then, we further perform the feature extraction and feature scaling on the skeleton information, to help the model learn more effectively. Regarding the classification model, since the fall events need to be dealt with urgently, we use the machine learning approach to classify instead of the deep learning approach. In the decision part, when the classification model thinks that a fall event has occurred, an alarm will sound. Otherwise, if everything is considered normal, it will continue to identify the next frame.

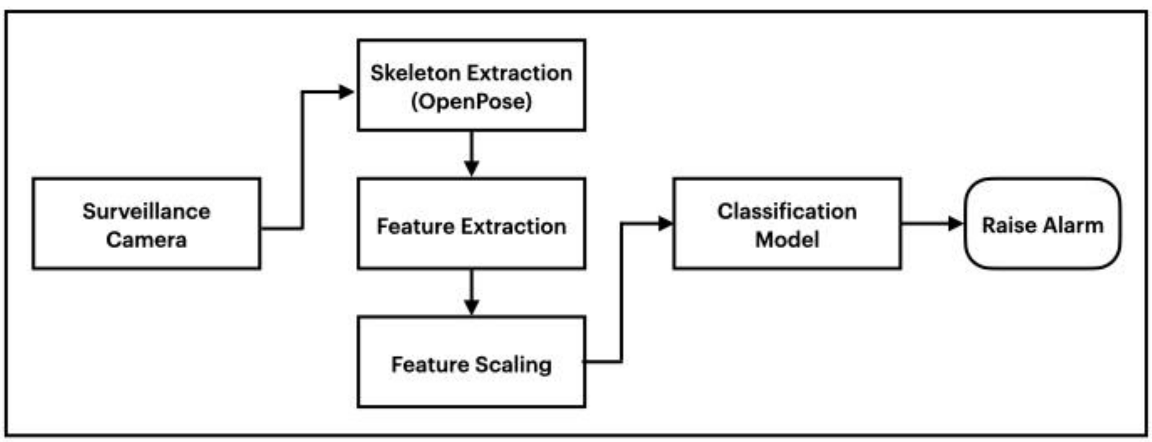


Figure 5.1: Fall Detection System Architecture

5.2 Preprocessing and Feature Extraction

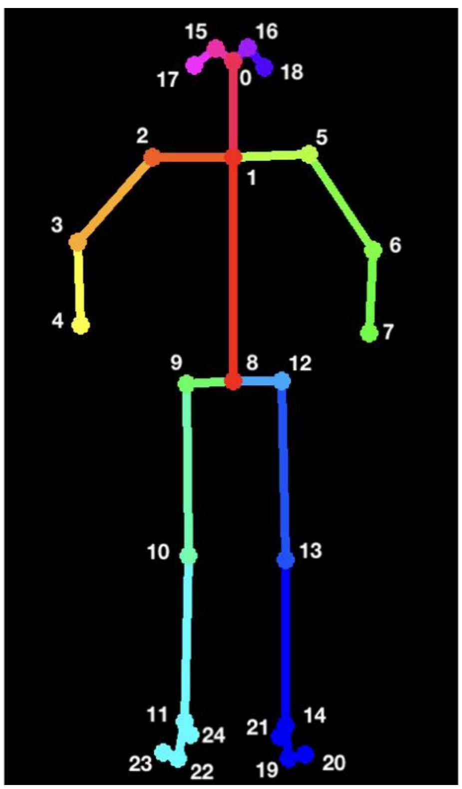
In most machine learning tasks, data preprocessing and feature extraction are the most important stages. The importance of features are more significant than model selection. Therefore, in the preprocessing part, we use pose estimation to extract skeleton information.

In some vision-based research, they use Microsoft Kinect to extract the skeleton feature, but the detecting distance is one of the drawbacks for Kinect. Therefore, we use a deep learning-based method called OpenPose as our pose estimation method. OpenPose is robust in multiple people scenarios. Since the skeleton size may be different in a variety of distinct distances, we further extract the key features from skeleton information to minimize the effect of distance.

5.2.1 Pose Estimation – OpenPose

To extract the skeleton information from RGB images, we use OpenPose as our pose estimation method, an open-source library developed by Cao et al. [26][62], from Carnegie Mellon University (CMU). OpenPose is a 2D pose estimation method that can effectively detect 25 body parts and form a body skeleton. OpenPose is the bottom-up method that detects the body part first and then forms the skeleton in terms of time performance. It performs well in environments with many or few people, or even different light conditions. Therefore, the runtime performance does not interfere with the increase of people. The most important thing is that compared with Microsoft Kinect, OpenPose's detection distance is not limited, which meets our fall detection system requirement very well.

Figure 5.2: OpenPose in Multiple People Scenario [33]



Number	Body Part
0	Nose
1	Neck
2	R Shoulder
3	R Elbow
4	R Wrist
5	L Shoulder
6	L Elbow
7	L Wrist
8	Mid Hip
9	R Hip
10	R Knee
11	R Ankle
12	L Hip
13	L Knee
14	L Ankle
15	R Eye
16	L Eye
17	R Ear
18	L Ear
19	L BigToe
20	L SmallToe
21	L Heel
22	R BigToe
23	R SmallToe
24	R Heel

Figure 5.3: OpenPose Keypoint Diagram and Keypoint Table [26]

In our approach, we only use 0-14 body keypoints to represent the human body. Furthermore, we exclude 15-24 body keypoints because even if these points are excluded, we can still see whether the body tends to fall. In this case, we consider these keypoints to be unrelated to a fall event.

5.2.2 Feature Extraction

Physical characteristics are significant in the past fall events, such as body deflection and acceleration during the fall. Therefore, besides extracting the skeleton feature, we further extract the 14 physical features from the skeleton information. By doing a further feature extraction, the model can learn the difference more effectively during the training, and the whole prediction becomes more interpretable. Next, we construct the features, including ratio feature, distance feature, acceleration feature, and deflection feature. Ratio features can represent the body outline. Distance and acceleration features can represent the movement of some important body parts frame by frame. Deflection features can explain more detailed body part status.

Ratio Feature:

When the Fall events occur, the most apparent feature is the change in the posture of the human body. In most ADL events, the human body skeleton's height is larger than the width. However, when the fall events occur, the skeleton's height tends to decrease, and the skeleton's width increases. Thus, we use the Height and Width ratio (HW ratio) as the feature to represent the body outline. Nonetheless, when the direction of the fall faces the camera, the width may not increase. Therefore, we refer to the 'Spine ratio' from Han et al. [30] to supplement the 'HW ratio'. For example, the length from keypoint 1 to keypoint 8 is Spine-length, and the length from keypoint 9 to keypoint 12 is Waist-length. Therefore, we use Spine length divided by Waist-length as the Spine ratio. In this way, because the

Spine-length decreases and the Waist-length stays stable when the fall direction faces the camera, the 'Spine ratio' decreases. Thus, the 'Spine ratio' can be used as a feature to identify the fall.

The detailed calculation process is as follows.

$$HW\ Ratio = \frac{Height}{Width} \ldots\ldots\ldots\ldots\ldots\ldots (5.1)$$

$$Spine\ Ratio = \frac{|Spine\ Length|}{|Waist\ Length|} \ldots\ldots\ldots (5.2)$$

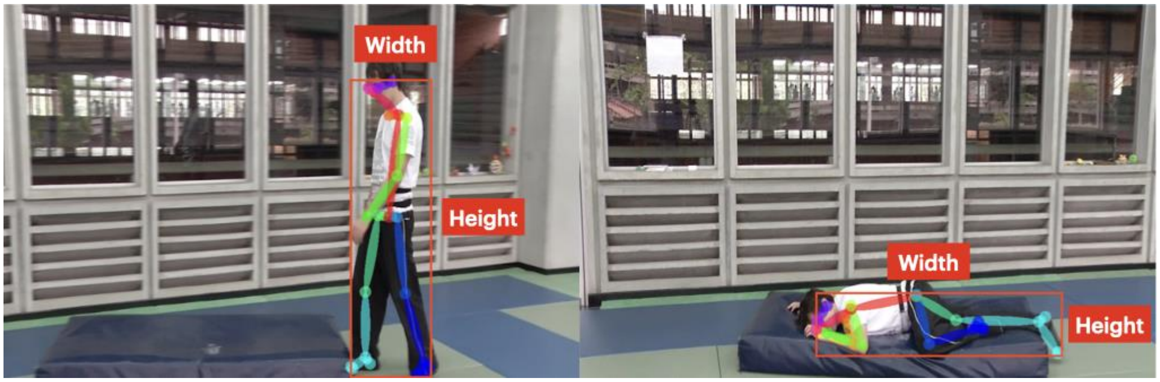


Figure 5.4: Simple illustration of HW Ratio [23]

Figure 5.5: Falling direction faces the camera [21]

In Figure 5.3, the 'HW ratio' changes rapidly when the Fall event occurs. In Figure 5.4, although the test person falls facing the camera, and the 'HW ratio' does not drop significantly, the 'Spine ratio' drops. From right to left, the 'Spine ratio' is 2.24, 2.09, 1.35, and 1.01, respectively.

Distance Feature:

There is usually a significant difference in height between the head, hips, and feet in everyday movement. However, when fall events occur, the vertical distance between the hips, head, and the ground, decreases significantly. In some cases, the head and hips even parallel to the feet. Therefore, we use the midpoint of keypoint 11, 14 to represent feet location. Then we calculate the vertical distance from the keypoint 0 (Neck) to the feet and the vertical distance from the keypoint 8 (Mid Hip) to the feet as features to further identify the body's current status.

The detailed calculation process is as follows.

$$\text{Neck to Feet Distance} = (y\ of\ Feet) - (y\ of\ Neck) \ \ldots.. \ (5.3)$$

$$\text{Hip to Feet Distance} = (y\ of\ Feet) - (y\ of\ Hip) \ \ldots\ldots.. \ (5.4)$$

Acceleration Feature:

In wearable sensor-based fall detection, the accelerometer is an effective indicator. The occurrence of fall events often causes a dramatic acceleration change in a short time. Therefore, we extract the acceleration feature from the skeleton information. In the research of Kangas et al. [28], they put the accelerometer in multiple body parts and tested the performance for fall detection. In their experiment, the position of the head and waist has the highest sensitivity and specificity. Thus, in our approach, we calculate the acceleration of keypoint 0, 1, 8 (Nose, Neck, Center of Waist). We only calculate the change of negative, vertical acceleration.

The detailed calculation process is as follows.

$$\text{Head Acceleration} = \frac{(y\ of\ Head) - (y\ of\ preHead)}{5} \ \ldots\ldots.. \ (5.5)$$

$$\text{Neck Acceleration} = \frac{(y\ of\ Neck)-(y\ of\ preNeck)}{5} \ldots\ldots\ldots (5.6)$$

$$\text{Hip Acceleration} = \frac{(y\ of\ Hip)-(y\ of\ preHip)}{5} \ldots\ldots\ldots\ldots\ldots (5.7)$$

Deflection Feature:

When a fall event occurs, the angle between the body and the ground usually changes greatly, in addition to the change in the body's contour. Therefore, we used the deflection angle proposed by Han et al. [30] to capture the deflection feature. To calculate the deflection angle of each body part, we use the gravity vector and 6 body vectors. The gravity vector is any vector parallel to the y axis. Body vector is a vector formed by two keypoints, spine vector is keypoint 1 to 8, waist vector is keypoint 9 to 12, Right Left (RL) calf vector is keypoint 10 to 11 and keypoint 13 to 14, and RL thigh vector is keypoint 9 to 10 and keypoint 12 to 13. Using the cosine function, we can find the angle between the body part and the ground, as shown in Figure 5.5.

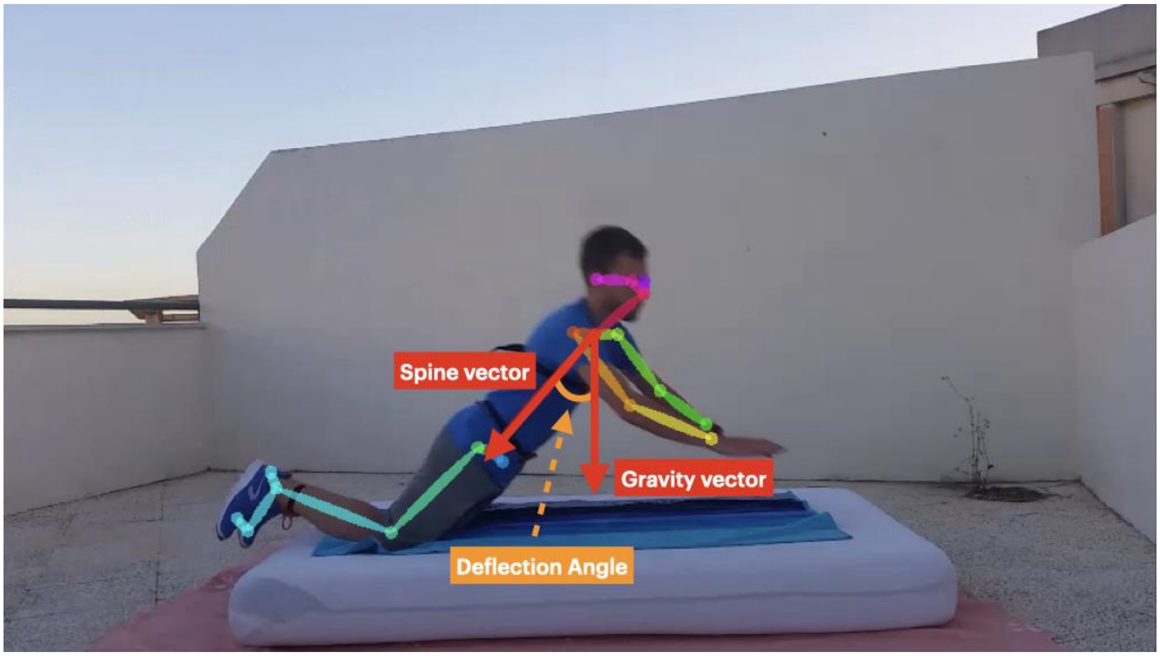


Figure 5.6: Spine Deflection Angle [22] [30]

The detailed calculation process is as follows.

$$\text{Deflection Angle} = \cos^{-1}\left(\frac{\overrightarrow{Body\ Vector} \cdot \overrightarrow{Gravity\ Vector}}{|\overrightarrow{Body\ Vector}| \times |\overrightarrow{Gravity\ Vector}|}\right) \ldots\ldots\ldots.. (5.8)$$

In addition to the above vector, we also measure the whole body tilt angle. We calculate the angles of the neck point with mid-hip, mid knees, and mid ankles, respectively. Finally, we choose the smallest angle as the body tilt angle. As shown in Figure 5.7, since the mid ankles can form the smallest angle, we use this angle to represent the body's tilt angle.

The detailed calculation process is as follows.

$$\text{Body Tilt Angle} = \tan^{-1}\left(\frac{y\ of\ Neck - y\ of\ BodyPart}{x\ of\ Neck - x\ of\ BodyPart}\right) \ldots\ldots\ldots\ldots (5.9)$$

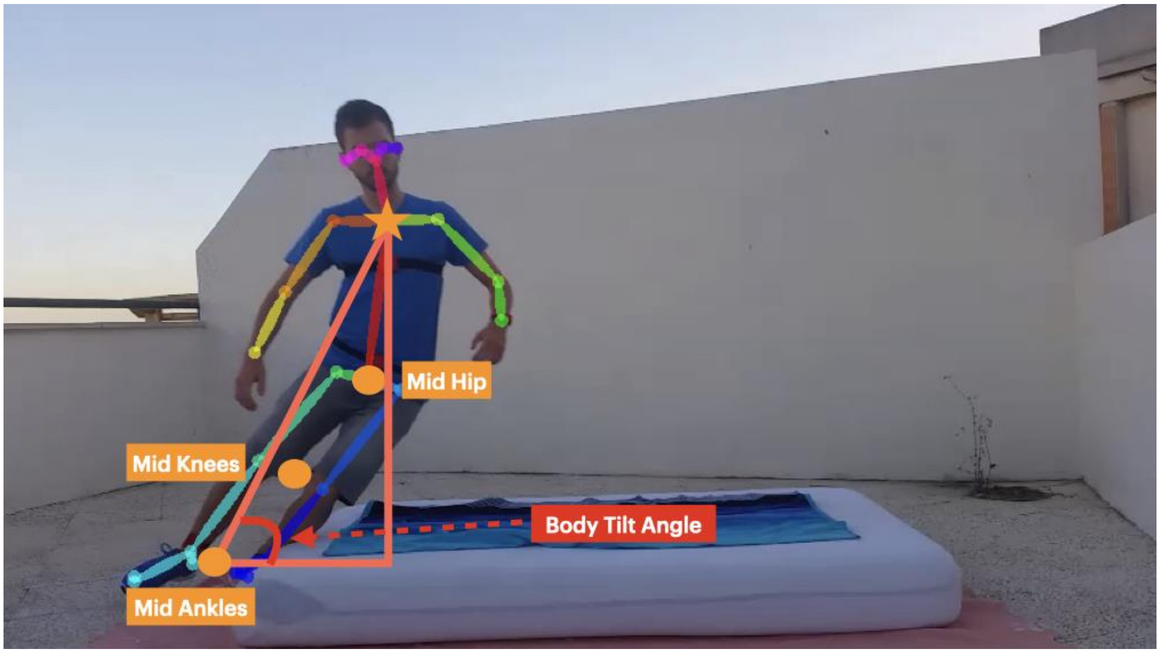


Figure 5.7: Body Tilt Angle [22] [30]

5.3 Imbalanced Data Handling

Data imbalance is a common problem in machine learning tasks. Most machine learning algorithms assume that the data is uniformly distributed. However, in a real-world problem, the data is often unevenly distributed. When the number of data is extremely different, the minority of classes are not represented in the model. This results in the majority of the class dominating the learning performance. To solve the imbalanced data problem, we use standard methods, including Sampling Methods and Anomaly Detection.

5.3.1 Sampling Methods

Sampling Methods are mainly used to balance data distribution and can be divided into Oversampling and Undersampling. Oversampling can increase the size of rare samples to achieve a balance. Oversampling can also be divided into Random Sampling and Synthetic Sampling. The data in Random Sampling may repeatedly appear, so that it may cause overfitting. Synthetic Sampling can use existing data to generate more samples, thereby avoiding overfitting. Undersampling can reduce the number of samples to achieve a balance. But the disadvantage is that the data is not complete, so the model can only learn a part of the whole. Thus, we usually use undersampling when all the class data are insufficient. Although our data is imbalanced, the number of each class is sufficient, therefore we did not use undersampling methods. In our approach, we use one kind of Random Sampling and three kinds of Synthetic Sampling, SMOTE, SMOTE Tomek, and ADASYN.

Four different approaches have different methodologies for sampling the data. Random sampling can duplicate some original data of minority classes to balance the data distribution. In particular, an issue that needs to be noticed is the fact that this approach causes overfitting easily. SMOTE starts by searching k nearest neighbors. It calculates the distance between k nearest neighbors to generate the new data. The data that are generated are relatively close in the feature space. SMOTE Tomek is a modification of SMOTE. It can generate the new data from the minority class and remove the data close to the majority class. ADASYN has an adaptive nature. It can generate hard-to-learn data, such as the data around the border, instead of data located in high-density areas.

5.3.2 Anomaly Detection

Isolation Forest:

Isolation forest is an unsupervised and non-parametric method that is suitable for continuous, numerical data. Isolation forest assumes that the outlier is sparsely distributed and far from the high-density data group. The idea of the method is that if the data is normal, you need more decision trees to separate the data. Conversely, if the data is abnormal, you can separate the data with fewer decision trees.

One Class SVM:

One Class SVM is an unsupervised method. It only uses one class of data to train the model. By utilizing the majority class to train the model, the model can learn a decision boundary and use that boundary to determine whether the new data is similar to the training data. If the boundary is exceeded, it is regarded as an anomaly. The kernel function we used is

Radial Basis Function (RBF), which can effectively project features to high dimensions, and make data have a good aggregation. Thus, this method usually performs well when the data dimension is high.

Elliptic Envelope:

Elliptic Envelope is an unsupervised algorithm. This algorithm assumes that the distribution of the data conforms to the Gaussian distribution. By estimating the covariance, this method encloses the data in an oval area. Any data outside this area will be identified as an outlier. It performs well when the data conforms to the Gaussian distribution.

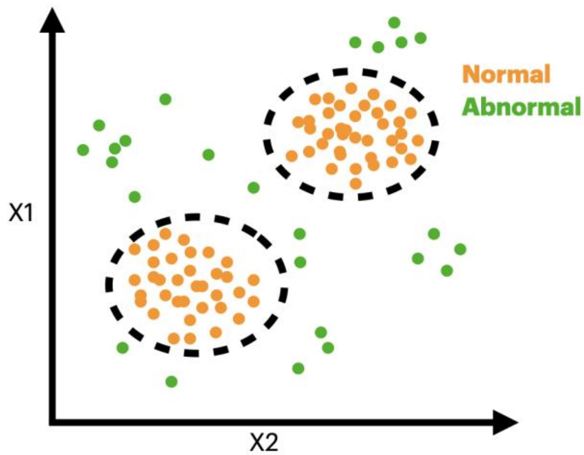


Figure 5.8: Simple Illustration of Anomaly Detection

5.4 Classfication Model

K-Nearest Neighbours:

K-nearest Neighbours (KNN) is a well-known pattern classification algorithm. It is one of the mature, and straightforward supervised machine learning algorithms. It is a method of classification based on the local distance feature. KNN is used on both classification and regression problems. In regards to classification problems, KNN begins with calculating the distance between a predicted datapoint. Then, it surrounds the data points and collects the closest data points of “k.”. Second, KNN determines the class based on the most common class in the “k” closest data points. In the regression problem, the predicted value is the average value of the “k” closest data points. K value is typically small.

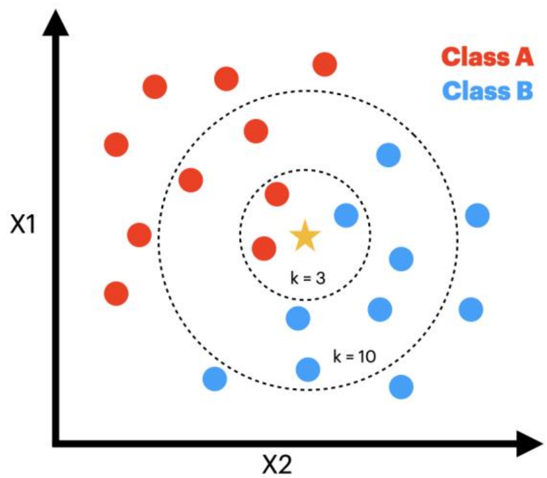


Figure 5.9: Simple Illustration of KNN

In Figure 5, the most common class in the 3 closest data points is Class A, when K = 3. The most common class in the 10 closest data points is Class B, when K = 10. Thus, the predicted class can be different, depending on the K value.

Support Vector Machines:

Support Vector Machines (SVMs) is a supervised machine learning algorithm, and it is widely used in industrial applications. SVMs use training data to find a decision boundary, called an optimal hyperplane. The optimal hyperplane separates the different classes with a possible wide gap, known as the highest margins. In addition to linear analysis cases, the SVMs are well-known in linearly inseparable cases. The SVMs' kernel functions can effectively transform the inseparable features from a low dimension to a high dimension, where it becomes easier to separate with a hyperplane. Due to the optimal hyperplane, the SVMs have robustness on sparse data.

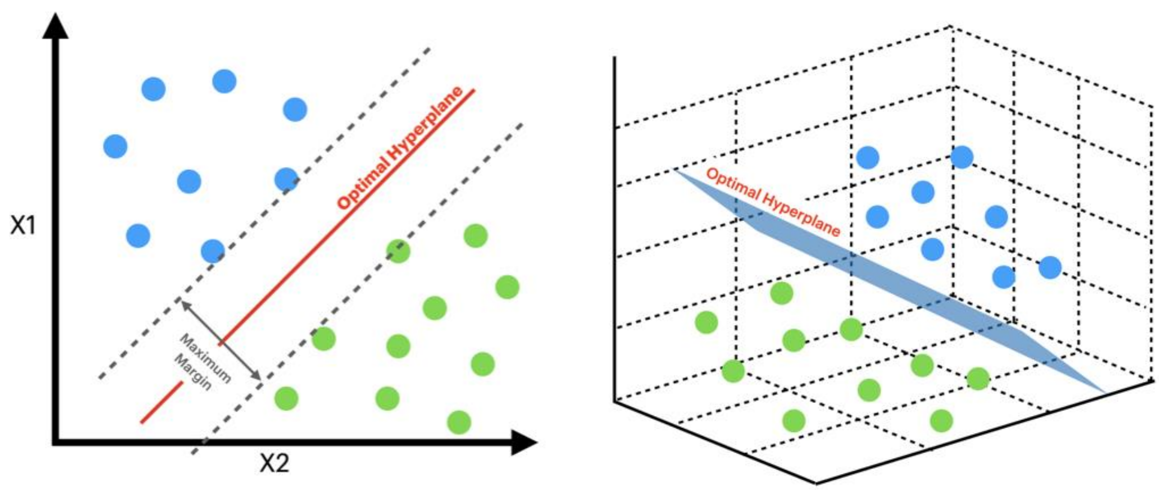


Figure 5.10: SVMs in 2 Dimensional and 3 Dimensional Spaces

Boosting:

Boosting is an ensemble machine learning algorithm that is an effective and widely used supervised learning method. It iteratively reweights the training data and trains the weak learners. Also, it finally consists of all the weak learners into strong learners. The false predicted data can gain more weight in the following training so that the next weak learner can improve the previous weak learner. Therefore, the boosting method is considered to be a practical approach when underfitting happens. In our approach, we use two boosting methods, AdaBoost and XGBoost [29].

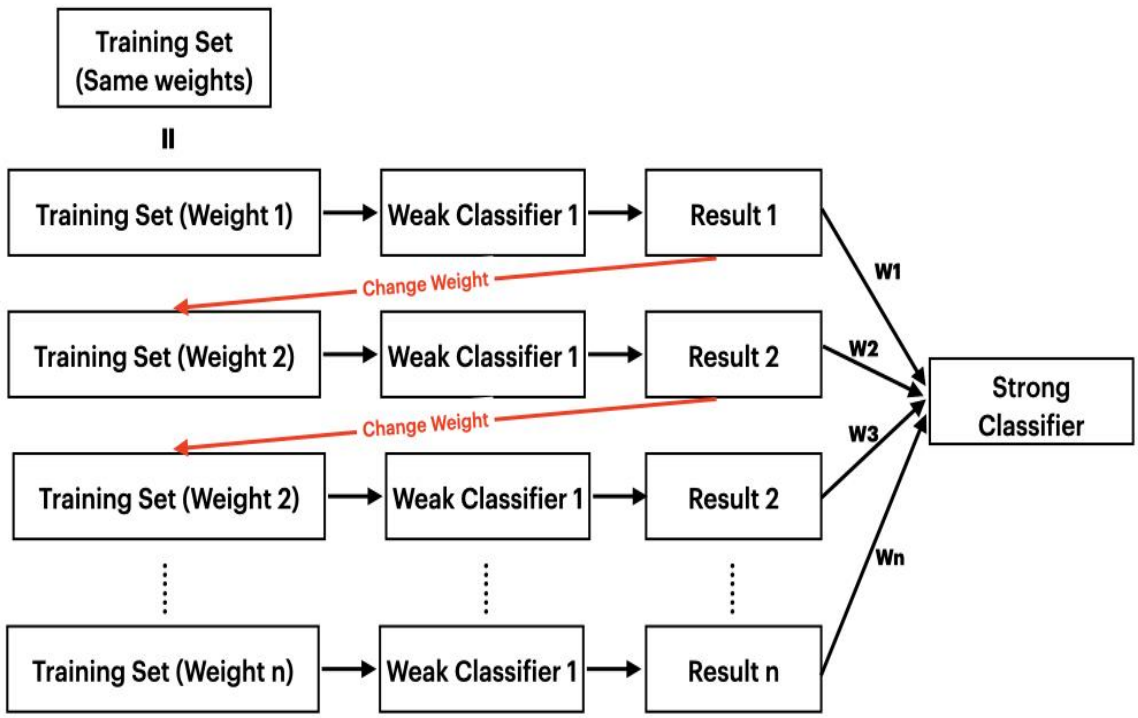


Figure 5.11: Simple Illustration of Boosting [32]

Adaptive Boosting (AdaBoost) is used with various machine learning algorithms to improve accuracy. It weights the training sample and trains multiple weak classifiers to compose one strong classifier, though the weak classifier can only become a bit better than random guessing. First, if we have N numbers of data samples in the weighting process, we equalize each data sample's weight to 1/N. After the training for the first weak classifier, the accurate predicted data sample's weight is deducted, and the false predicted data sample's weight is increased. Second, the next weak classifier is trained with reweighted data samples, focusing on hard-to-classify data and improving accuracy. We are finally composing all the weak classifiers with specific weight to one strong classifier. AdaBoost is a high-precision method, and it is not easy to overfitting.

XGBoost [29] is a Tree ensemble model. XGBoost uses additive training to preserve the model and attach a new tree in each iteration, to improve the previous tree. Comparing to AdaBoost, it uses weight to strengthen the hard-to-classify data. On the other hand, XGBoost uses residual to improve the accuracy. Although both algorithms follow the Boosting concept, XGBoost has made significant improvements in algorithm optimization and system optimization, so it has robust scalability, speed, and accuracy. Moreover, XGBoost is good at handling missing values and has features of automatic feature selection. As a result, it is a popular and widely used method in most data science projects and Kaggle competitions.

Chapter 6. Experimental Results

6.1 Experiment Configuration

We implemented the platform in Ubuntu 18.04. Python 3.8 is the main coding language, and the developing IDE is Jupiter notebook. Machine learning, sampling method, and anomaly detection are based on the scikit-learn package. Pose estimation is based on the OpenPose package. We use OpenCV to display and process every image. Since OpenPose needs GPU resources to speed up the image processing time, we use the Tesla P100-PCIE-16GB for our GPU resources.

More detailed configurations are shown in Table 6.1.

Experiment Configuration	
System	Ubuntu 18.04
CPU	Intel(R) Xeon(R) Silver 4114 CPU @ 2.20GHz
GPU	Tesla P100-PCIE-16GB
RAM	16GB
Cuda version	11.1.1
Cudnn version	8.0.5.39

Table 6.1: Experiment Configuration

6.2 Dataset

6.2.1 Data Labeling

To train our model, we manually labelled the data that we collected. We divided the data into three classes, 'Normal', 'Fall', and 'Lying'.

The definition of 'Lying' indicates a person's posture is settling on the ground or lying on the bed. We all identify as 'Lying' class. The definition of 'Fall' refers to the interval between 'Normal' status that changes to 'Lying' status. Regardless of whether the action taken is sitting or walking, this interval belongs to the 'Fall' class. This is only as long as the person ends up lying on the ground. We do not have many restrictions for the 'Normal' class. Any status that does not fall into 'Lying' and 'Fall,' belongs to the 'Normal' class. Therefore, some easily misidentified actions, such as squatting, sitting, and jumping, belong to the scope of "Normal."

After the data labelling, the 'Normal' class has 280770 frames, the 'Fall' class has 8126 frames, and the 'Lying' class has 20841 frames. Thus, the data is highly imbalanced. Table 6.1 is the result of the data proportion.

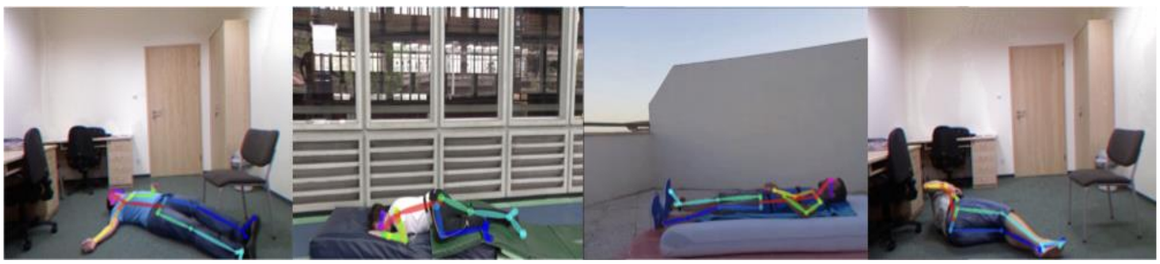


Figure 6.1: Example of 'Lying' class [21,22,23]

Figure 6.2: Example of 'Falling' class [21,22,23]

Figure 6.3: Example of 'Normal' class [21]

	Normal	Fall	Lying
Frames	280770	8126	20841
Percentage	90.65%	2.62%	6.73%

Table 6.2: Data Proportion.

6.2.2 Data Preprocessing

In some cases, when body occlusions happen, OpenPose cannot effectively detect every key point of the body, which causes the missing value of the feature, and outlier in ratio features. To exclude the outlier in 'HW ratio' and 'Spine ratio', we only keep the ratio data in the 10-90 percentile range, because the value is more reasonable. To make the model understand whether the feature is missing, we add new columns to the corresponding feature to indicate whether the feature is lost. For example, if the 'Spine ratio' is missing, the corresponding new value of 'Have_Spine_ratio' is 0. On the other hand, if the 'Spine ratio' is not missing, the value of 'Have_Spine_ratio' is 1. We do not have corresponding columns for 'HW ratio,' 'Head Acc,' 'Neck Acc,' and 'Spine Acc' because those features have no missing values. The last step is to fill in the missing value. We evaluate mode, median, and mean values to decide the replacement value. The mean value has the best performance in section 6 of the experiment, so we replace the missing value with the mean value. We replace the value according to the data's label. If the label is 'Normal', we replace the missing value with the mean value of the 'Normal' class, and so on. Figure 6.5 is the illustration of our missing value handling approach.

Figure 6.4: OpenPose misses detecting body keypoints when occlusion happens. [21]

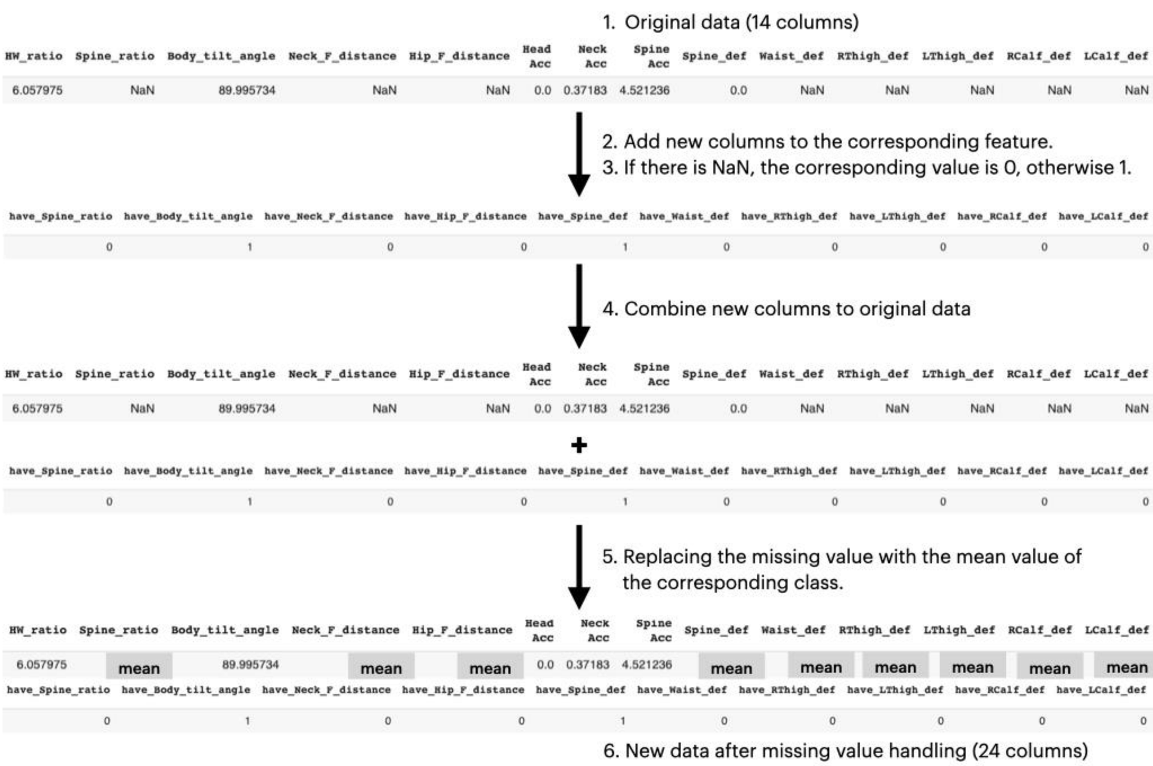


Figure 6.5: Missing Value Handling Illustration

6.2.3 Split Training Data and Testing Data

To evaluate whether our model is overfitting or not, we use 80% of data for training and 20% of data for testing. Thus, we have three different experiments. First, we trained the model directly with training data to assess the model's performance on imbalanced data. Second, we use the oversampling method on training data, in order to balance out the number of data in each class. Finally, we evaluate the performance with testing data that still stays imbalanced.

6.3 Evaluation Metrics

A good evaluation metric can evaluate the model's performance effectively. However, when the data is skewed, the standard evaluation metrics, such as accuracy and error, can lead to misleading information. For example, suppose we have 5 positive data and 95 negative data. In that case, the model will receive a 95% accuracy rate when predicting every data to negative data, regardless of whether it can predict any positive data. Therefore, in the unbalanced dataset. It is more important to predict the minority of data, successfully. Based on Vallabh et al. [31] 's fall detection review, we evaluate our approach via Precision, Recall, F1 Score, Specificity, and Area Under the Curve (AUC) of Receiver Operating Characteristic (ROC), which are commonly used in previous Fall detection research.

Most of the values can be calculated by the following outcomes. True Positive (TP), the model predicts Fall events to occur, and Fall events occur. False Positive (FP), The model predicts Fall events to occur, and no Fall events occur. True Negative (TN), the model predicts no Fall events to occur, and no Fall events occur. False Negative (FN), the model predicts no Fall events to occur, but Fall events occur.

$$Precision = \frac{TP}{TP+FP} \ldots\ldots\ldots\ldots\ldots \quad (6.1)$$

$$Recall = \frac{TP}{TP+FN} \ldots\ldots\ldots\ldots\ldots\ldots.. \quad (6.2)$$

Precision can represent how much of the data is predicted to Fall, as they eventually lead to a Fall. Recall can represent how much the model can predict the actual Fall. Thus, if we want the model to identify every possible Fall event, the higher the Recall rate, the better.

$$F1\ Score = \frac{2\times(Precision\times Recall)}{Precision+Recall} \quad \ldots\ldots\ldots (6.3)$$

$$Specificity = \frac{TN}{FP+TN} \quad \ldots\ldots\ldots\ldots\ldots\ldots (6.4)$$

F1-Score is the combination of Precision and Recall. The Specificity can represent how much the model can predict actual ADL events. Thus, we can check the Recall and Specificity to make sure the model learns all the different label characteristics.

Receiver Operating Characteristic (ROC) curve is the combination of Recall and Specificity. The value of AUC is the area below the ROC curve. Generally, the AUC score is between 0.5 and 1. The larger AUC, the better the classification performance.

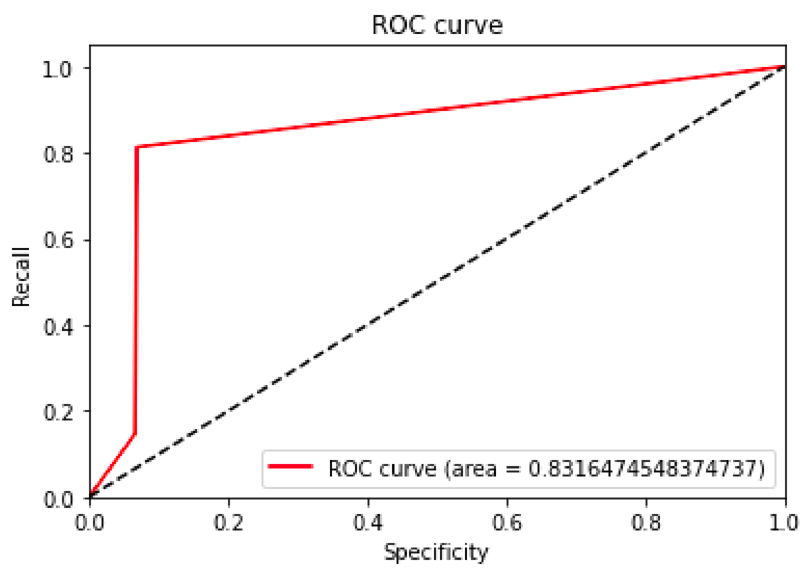


Figure 6.6: ROC Curve and AUC score

6.4 Analysis of Experiment Result

6.4.1 Machine Learning without Oversampling

In the beginning, we test the performance of each model on an imbalanced dataset. In the proportion of our dataset, 'Normal' accounts for 91%, 'Fall' accounts for 2.6%, and 'Lying' accounts for 6.4%. Training data and testing data maintain the same ratio.

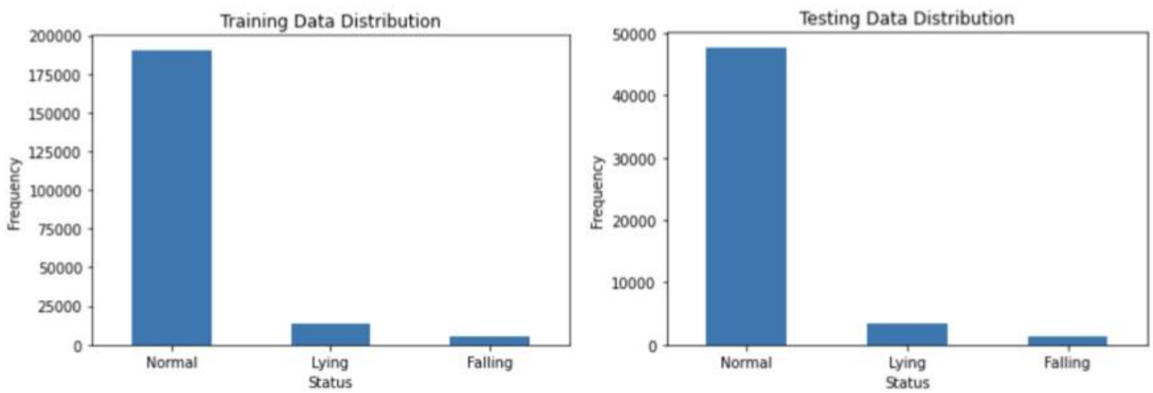


Figure 6.7: Data Distribution (without Oversampling)

In the experiment result, each model has good classification performance on the 'Normal' class, and both Precision and Recall are above 0.99. 'Lying' class has the second-best performance. For Precision and Recall, all models can achieve above 0.90. Although 'Lying' accounts for only 6.4%, 'Lying' characteristics are quite different from the other classes, so even if the data is less, the model can still learn the characteristic of 'Lying'. As for the 'Fall' class, since there is less data, most models' perform poorly, other classes. For example, SVMs and XGBoost have F1 scores lower than 0.7, and XGBoost and KNN's F1 scores are 0.74 and 0.82. The possible reason that SVMs perform poorly is that the sequence of 'Normal,' 'Fall,' and 'Lying' is a consecutive movement. This means that the data acquires many overlaps amongst each other. Therefore, the SVMs cannot effectively find the best hyperplane to divide each class. On the other hand, KNN has the best performance in every class, surpassing the common machine learning competition algorithm, XGBoost. The possible reason is that KNN only calculates the nearest neighbors to make a prediction, so the computation is local, and the effect from the imbalanced data is less.

Normal	Precision	Recall	Specificity	F1-Score	AUC
KNN	1.00	1.00	0.97	1.00	0.98
AdaBoost	0.99	0.99	0.95	0.99	0.97
XGBoost	0.99	1.00	0.92	1.00	0.96
SVMs	1.00	0.98	0.98	0.99	0.97
Fall	Precision	Recall	Specificity	F1-Score	AUC
KNN	0.92	0.73	1.00	0.82	0.86
AdaBoost	0.65	0.67	0.99	0.66	0.82
XGBoost	0.91	0.63	1.00	0.74	0.81
SVMs	0.51	0.81	0.98	0.62	0.89
Lying	Precision	Recall	Specificity	F1-Score	AUC
KNN	0.94	0.98	0.99	0.96	0.98
AdaBoost	0.92	0.92	0.99	0.92	0.95
XGBoost	0.94	0.97	1.00	0.96	0.98
SVMs	0.95	0.95	1.00	0.95	0.97

Table 6.3: Different machine learning model performance

Model	Parameters
KNN	n_neighbors=3, weight='distance', p=1
AdaBoost	algorithm='SAMME.R', learning_rate=1, n_estimator=50
XGBoost	Learning_rate=0.01, n_estimators=1000, max_depth=6, subsample=0.8, colsample_bytree=1, gamma=1
SVMs	kernel='rbf', class_weight='balanced', C=50

Table 6.4: Parameters for Machine Learning models

6.4.2 Machine Learning with Oversampling

Oversampling is an effective way to deal with imbalanced data. It can generate new data, via random sampling or data synthesis. Since we have the best performance on KNN in the previous experiment, we use different oversampling methods to balance our data and evaluate the performance, via KNN. First, we divide the data into training data and testing data. Then, we perform oversampling on training data. Therefore, the proportion of training data becomes balanced data, while testing data retains the original imbalanced distribution.

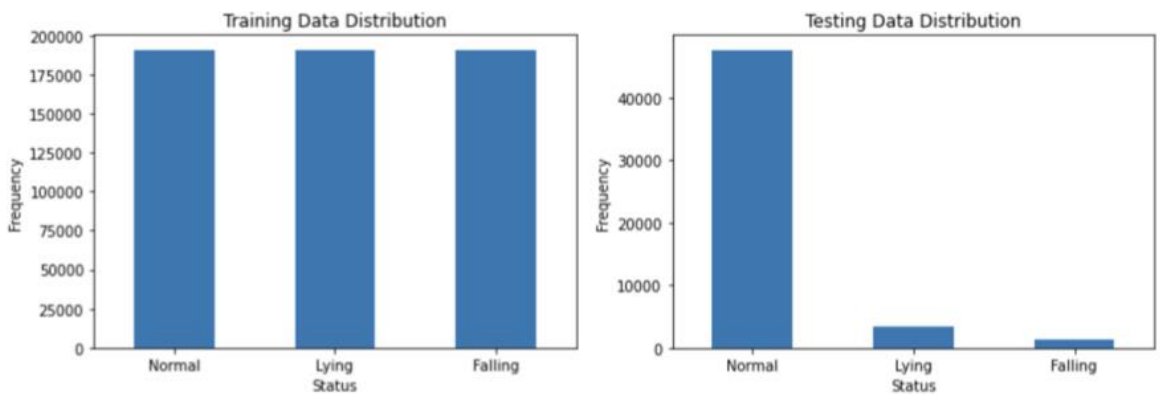


Figure 6.8: Data Distribution (Oversampling)

After oversampling, we use the testing data to test whether the model has overfitting. The experiment shows that both 'Normal' and 'Lying' classes have good performance. Furthermore, the Recall of the 'Fall' class improves from 0.73 to 0.88, which means the model can detect 88% of real fall events. Finally, comparing different oversampling methods, SMOTE Tomek and SMOTE have the best Recall, and Random Sampling has the most favourable Precision. To consider both Precision and Recall, we can refer to the F1 score. Random Sampling has the highest F1 score, 0.85. In conclusion, four methods effectively enhance the model's learning of a minority class, and there is not much difference in the performance.

Normal	Precision	Recall	Specificity	F1-Score	AUC
Random Sampling	1.00	1.00	0.99	1.00	0.99
SMOTE	0.99	0.99	0.94	0.99	0.96
SMOTE Tomek	1.00	0.99	0.99	1.00	0.99
ADASYN	1.00	0.99	0.99	1.00	0.99
Fall	Precision	Recall	Specificity	F1-Score	AUC
Random Sampling	0.87	0.84	1.00	0.85	0.92
SMOTE	0.79	0.88	0.99	0.83	0.93
SMOTE Tomek	0.79	0.88	0.99	0.83	0.93
ADASYN	0.75	0.87	0.99	0.81	0.93
Lying	Precision	Recall	Specificity	F1-Score	AUC
Random Sampling	0.95	0.98	1.00	0.96	0.98
SMOTE	0.96	0.97	1.00	0.97	0.98
SMOTE Tomek	0.96	0.97	1.00	0.97	0.98
ADASYN	0.95	0.97	1.00	0.96	0.98

Table 6.5: Different oversampling methods performance

6.4.3 Anomaly Detection

Another method to deal with imbalanced data is to identify data, via anomaly detection. Since anomaly detection is a binary classification, we merged the "Fall" and "Lying" data into the "Abnormal" class. Using anomaly detection, we consider the minority class as outliers. Thus, "Normal" data accounts for 91% of the data distribution, and 'Abnormal' data accounts for 9%.

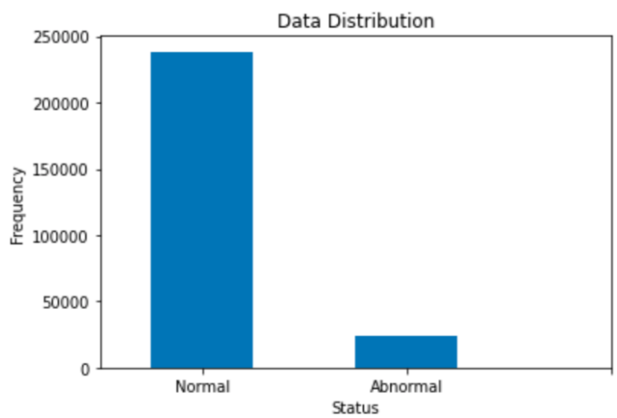


Figure 6.9: Data Distribution (Normal and Abnormal)

In the experiment results, the Elliptic Envelope has the best performance. However, most of the model's precisions are low. The possible reason is that the model predicts more normal data as abnormal, which causes the Recall rise, but the Precision drop. Thus, evaluating the performance with F1-Score, the machine learning-based methods perform better than anomaly detection-based methods. The possible reason is that there is a lot of overlap in the dataset, so the anomaly detection method cannot separate 'Normal' and 'Abnormal' well.

	Precision	Recall	Specificity	F1-Score	AUC
IF	0.59	0.65	0.95	0.62	0.79
EE	0.67	0.75	0.96	0.71	0.85
OCSVM	0.50	0.25	0.97	0.33	0.60

Table 6.6: Different anomaly detection methods performance on "Abnormal" class

6.5 Performance on Image:

After the previous experiments, we use the KNN with the Synthetic Minority Oversampling Technique (SMOTE) Tomek, to test the performance on the image. In Figure 6.10, results show that our approach can successfully identify the different scenarios. Furthermore, although OpenPose sometimes misses detecting the body keypoints, our approach can still make a prediction based on the remaining body keypoints.

Figure 6.10: Performance on single-person scenarios [21,22,23]

Moreover, our methods not only can work on single-person scenarios, but also multi-people scenarios. In multi-people scenarios, we test our approach with IASLAB-RGBD Fallen Person Dataset [35]. The images are taken in the lab environment, and each image has more than one person. Although we never train those images with our model, our model can still identify the fall events in the image.

Figure 6.11: Performance on multi-people scenarios [35]

Since we extract the interpretable features from the skeleton information, we can show the key features on the images to have more information of when abnormal events happen.

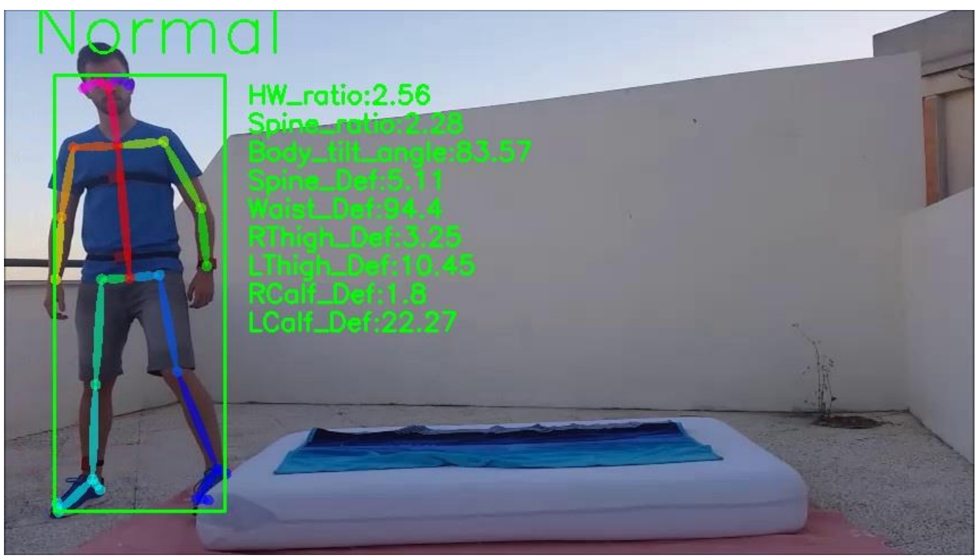


Figure 6.12: Normal event with key information. [22]

Figure 6.13: Abnormal event with key information. [22]

Chapter 7. Discussion

7.1 Machine Learning Performance

From previous vision-based approaches, they mainly use threshold and machine learning approaches as classifiers. Machine learning is preferable since it can adjust the different shapes of an individual. Using traditional machine learning methods as classifiers, it can receive good performance in previous research. However, before performing oversampling methods, our machine learning approaches do not perform well in the experiment, except for KNN. The possible reason may be our dataset. We collect four different fall datasets and one gait dataset. The data proportion is 90.6% Normal, 2.6% Fall, and 6.8% Lying, which is highly imbalanced. Each dataset is recorded in different environments. Although this added more variation into the experiment, it is close to real-world situations, since fall events have lots of variation. After performing oversampling methods, the performance improves. Performance in test data shows that we do not have an overfitting problem. Performance in the video can identify fall events correctly, even with the fallen image from another dataset. However, it is important to note that SVMs shows good performance in previous research, but it has the worst performance in our experiment. The possible reason is that since fall is structured to move consecutively, each class has a large overlap with each other. This makes it difficult for SVMs to find the hyperplane.

7.2 Pros and cons of pose estimation-based approaches

A survey paper from Biswas et al. [45], summarized the main challenges in vision-based, fall detection approaches. The challenges are poor identification of pose, the area at home or public area, number of people in the frame, poor lighting conditions, occlusion, subject's distance, and usage of aid accessories [44]. Since we use OpenPose as our pose estimation method, OpenPose's performance significantly influences our approach. OpenPose can detect multiple people and even at different distances. If a person's lower body is occluded, our approach can make a detection based on upper body features. Thus, occlusion's problem is lighter than previous research. Using skeleton information is much interpretable and easier to understand, which is useful when cooperating with healthcare workers. The cons are the same as OpenPose. Some common failure cases are rare poses, overlapping with other people, and acquiring a false-positive on a statue or reflection. That false skeleton information causes a false alarm in our approach. The solution can be utilizing object detection techniques to check whether skeleton data is included in the bounding box.

7.3 Limitation

Fortunately, our approach can successfully detect falls, via skeleton information, but in regards to real-life applications, it still has a long way to go. There are some limitations to our approach.

The faster the treatment, the less chances of fatality. Fall detection systems should be able to detect falls immediately and call for help, as soon as possible. Since OpenPose and the classification model has a high computational demand, our approach cannot perform the real-time performance in our configuration setting. The higher-level computation resource is needed, to work in the real situation.

As for the dataset, the collected fall data are simulated data, which is not representative of the real fall event. Moreover, individuals in the dataset are not elderly people. In reality, the elderly may use tools, such as canes. We do not acquire that data in our dataset, so it is difficult to validate the effectiveness on the elderly, in a real-life setting.

OpenPose also has some limitations. Since OpenPose is a 2D pose estimation model, the skeleton information does not include depth information. Lack of depth information may cause the wrong identification of different postures, and make it difficult to utilize the same model in different camera angles. This is because the posture can have different shapes of the skeleton, in different angles. In some cases, a human's mirror reflection and posters with humans, can also be predicted as human objects, which may cause the wrong detection.

7.4 Recommendation for further research

Regarding further research, pose estimation-based approaches are a new trend, which has a lot of potential in future applications. Instead of using keypoint coordinates as features, we recommend extracting the features from skeleton information. This is more robust on different frame sizes. Since pose estimation is used a lot in motion recognition, training models with more activities can improve the robustness of the model. In our experiment, most of the data in the Normal class is gait data. Normal data also include a few activities such as sitting, squatting, etc. Adding more kinds of activities can allow the application to deal with different situations. Adding more abnormal events for detection can also be another direction since cameras can be used in multi-people scenes, the more functional, the better.

The Health science field is definitely a hot research field for further application. Due to the shortage of caregivers, ageing society is a problem that we cannot ignore. Health care costs are increasing in developed countries, such as the USA and Canada. Automation is the solution to this financial burden. Combining with health science and technique, more domain knowledge is required. Fall prevention is a more important challenge than fall detection. Most research concludes that falls are a mix-factor event. There are intrinsic and extrinsic factors that can strongly influence people's safety. Although the most decisive factor has not been realized, gait assessment is considered the most significant fall risk assessment. With the help of pose estimation-based approaches, automated fall risk assessment can be deployed everywhere with cameras. The tilt angle of the body, and the moving distance of the leg, can all be analyzed. Pose estimation can fill the research gap between fall detection and fall prevention.

The Robotics field is also a potential direction. The robotic technique is developing in industry and health science. Robotics and automated manufacturing are used in factories to enhance efficiency and save budget. Robotics can assist in surgery and diagnoses in health care. The robot mainly acts as an assistant nowadays. However, in the future, we expect robots to do more. We research human-robot interaction to add humanity to machines. Those humanity features can build trust and support systems with humans. We want a health assistive robot to take care of our family, support us physically, and mentally. Nowadays, we already have those robots in the market with human features [48][49][50][61]. Pose estimation approach can help robots understand human posture, and help to prevent accidents. From assistant to protector, robotics still has lots of potentials to explore.

Figure 7.1: Current assistive robots in the market, from left to right are, ElliQ, Mabu, Rudy, and Zenbo [48][49][50][61]

Chapter 8. Conclusions and Future Works

In this thesis, the problem we want to solve is fall risk detection. We discuss the factors for fall risk and the statistics in health care. Fall risk is dangerous to all individuals, especially for the elderly. The consequence of falls may cause physical injury and mental trauma. We need to take proper actions and steps to prevent it. Due to the shortage of health workers and the increasing financial burden on the health care system, automated fall risk detection can be the solution. We propose a new pose estimation-based fall detection algorithm, via RGB camera. We use OpenPose as a feature extractor to extract the skeleton data and then transfer them into 14 new features. New features are more interpretable compared to skeleton data. As for the dataset, we compose our dataset with four fall datasets and one gait dataset. The dataset is highly imbalanced, which meets real-world situations. In the experiment, we evaluate the performance of sampling methods and anomaly detection on imbalanced data. KNN plus oversampling has the best performance. Although OpenPose misses some body keypoints sometimes, our approach can base on the remaining feature to make a decision. And most importantly, compared to previous research on fall detection, our fall detection approach can handle multi-people scenarios.

In future works, we will change from detection to prevention. More data is needed to increase the diversity of Fall and ADL events. Surveying more domain knowledge in health science can help us decide on more crucial features. Since the camera has a wide range of views, we can add more abnormal events for the model to understand. To understand human activity, object detection combined with pose estimation can also be the next research direction. In another perspective, privacy issues are always the biggest drawbacks of vision-based approaches. It is acceptable to set up cameras in public areas, but people do not prefer setting up surveillance cameras in the home area, even though we only use skeleton information. Therefore, the household assistive robot is the alternative way. Nowadays, robotic products have started to apply in different fields, such as health care, industry, etc. Most robots contain multiple sensors, including cameras. Although we do not prefer to have surveillance cameras in homes, families and the elderly people have a more positive and acceptable attitude towards robots [47]. Some household robots can support medical assistance, such as medicine tracking and reminder [48][49][50]. During the pandemic period, using the robot with health care functions can prevent infection and lower the stress of health workers. Since the shortage of health workers and caregivers [51], adding more health care functions with the assistive robot is definitively the future direction.

Reference

[1] P. Canada, “Seniors' Falls in Canada: Second Report,” *Government of Canada*, April 10, 2014. [Online]. Available: https://www.canada.ca/en/public-health/services/health-promotion/aging-seniors/publications/publications-general-public/seniors-falls-canada-second-report.html

[2] S. E. Lamb, E. C. Jørstad-Stein, K. Hauer, and C. Becker, “Development of a common outcome data set for fall injury prevention trials: the Prevention of Falls Network Europe consensus,” *Journal of the American Geriatrics Society*, vol. 53, no. 9, 2005, pp. 1618-1622.

[3] A. F. Ambrose, G. Paul, and J. M. Hausdorff, “Risk factors for falls among older adults: a review of the literature,” *Maturitas*, vol. 75, no. 1, 2013, pp. 51-61.

[4] S. Deandrea, E. Lucenteforte, F. Bravi, R. Foschi, C. La Vecchia, and E. Negri, “Risk factors for falls in community-dwelling older people: a systematic review and meta-analysis," *Epidemiology*, 2010, pp. 658-668.

[5] S. L. Greenspan, E. R. Myers, D. P. Kiel, R. A. Parker, W. C. Hayes, and N. M. Resnick, “Fall direction, bone mineral density, and function: risk factors for hip fracture in frail nursing home elderly,” *The American journal of medicine*, vol. 104, no. 6, 1998, pp. 539-545.

[6] C. Smeesters, W. C. Hayes, and T. A. McMahon, “Disturbance type and gait speed affect fall direction and impact location,” *Journal of biomechanics*, vol. 34, no. 3, 2001, pp. 309-317.

[7] W. S. Leung, H. T. Chi, M. H. Hu, and M. T. Liu, “Fall mechanism and injury severity in community dwelling older people,” *Formosa J Phy Ther*, vol. 30, 2005, pp. 105-15.

[8] J. T. Perry, S. Kellog, S. M. Vaidya, J. H. Youn, H. Ali, and H. Sharif, “Survey and evaluation of real-time fall detection approaches,” *In 2009 6th International Symposium on High Capacity Optical Networks and Enabling Technologies (HONET)*, December 2009, pp. 158-164.

[9] M. Kangas, A. Konttila, P. Lindgren, I. Winblad, and T. Jämsä, “Comparison of low-complexity fall detection algorithms for body attached accelerometers,” *Gait & posture*, vol. 28, no. 2, 2008, pp. 285-291.

[10] S. S. Khan and J. Hoey, “Review of fall detection techniques: A data availability perspective,” *Medical engineering & physics*, vol. 39, 2017, pp. 12-22.

[11] W. Wu, S. Dasgupta, E. E. Ramirez, C. Peterson, and G. J. Norman, “Classification accuracies of physical activities using smartphone motion sensors,” *Journal of medical Internet research*, vol. 14, no.5, e130, 2012.

[12] F. Werner, J. Diermaier, S. Schmid, and P. Panek, “Fall detection with distributed floor-mounted accelerometers: An overview of the development and evaluation of a fall detection system within the project eHome,” *In 2011 5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops*, May 2011, pp. 354-361.

[13] M. Daher, A. Diab, M. E. B. El Najjar, M. A. Khalil, and F. Charpillet, “Elder tracking and fall detection system using smart tiles,” *IEEE Sensors Journal*, vol. 17, no. 2, 2016, pp. 469-479.

[14] M. Volkhardt, F. Schneemann, and H. M. Gross, “Fallen person detection for mobile robots using 3d depth data,” *In 2013 IEEE International Conference on Systems, Man, and Cybernetics*, October 2013, pp. 3573-3578.

[15] A. Yajai, A. Rodtook, K. Chinnasarn, and S. Rasmequan, "Fall detection using directional bounding box," *In 2015 12th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, July 2015, pp. 52-57.

[16] K. L. Lu and E. T. H. Chu, "An image-based fall detection system for the elderly," *Applied Sciences*, vol. 8, no. 10, 1995, 2018.

[17] W. Chen, Z. Jiang, H. Guo, and X. Ni, "Fall detection based on key points of human-skeleton using openpose," *Symmetry*, vol 12, no. 5, 744, 2020.

[18] G. Sun and Z. Wang, "Fall detection algorithm for the elderly based on human posture estimation," *In 2020 Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*, April 2020, pp. 172-176.

[19] Z. Huang, Y. Liu, Y. Fang, and B. K. Horn, "Video-based fall detection for seniors with human pose estimation," *In 2018 4th International Conference on Universal Village (UV)*, October 2018, pp. 1-4.

[20] S. Jeong, S. Kang, and I. Chun, "Human-skeleton based Fall-Detection Method using LSTM for Manufacturing Industries," *In 2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, June 2019, pp. 1-4.

[21] B. Kwolek and M. Kepski, "Human fall detection on embedded platform using depth maps and wireless accelerometer," *Computer methods and programs in biomedicine*, vol. 117, no. 3, 2014, pp. 489-501.

[22] E. Casilari, J. A. Santoyo-Ramón, and J. M. Cano-García, "Umafall: A multisensor dataset for the research on automatic fall detection," *Procedia Computer Science*, vol. 110, 2017, pp. 32-39.

[23] A. Sucerquia, J. D. López, and J. F. Vargas-Bonilla, “SisFall: A fall and movement dataset,” *Sensors*, vol. 17, no. 1, 198, 2017.

[24] E. Auvinet, C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, “Multiple cameras fall dataset,” *DIRO-Université de Montréal, Tech. Rep*, 2010, 1350.

[25] D. López-Fernández, F. J. Madrid-Cuevas, and Á. Carmona-Poyato, M. J. Marín-Jiménez, and R. Muñoz-Salinas, “The AVA multi-view dataset for gait recognition,” *In International workshop on activity monitoring by multiple distributed sensing*, Springer, Cham, August 2014, pp. 26-39.

[26] Z. Cao, T. Simon, S. E. Wei, and Y. Sheikh, “Realtime multi-person 2D pose estimation using part affinity fields,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[27] A.X. Araya and E. Iriarte, “Fear of Falling among Community-dwelling Sedentary and Active Older People,” *Invest Educ Enferm,* vol 39, no. 1, 2021.

[28] M. Kangas, A. Konttila, P. Lindgren, I. Winblad, and T. Jämsä, “Comparison of low-complexity fall detection algorithms for body attached accelerometer,” *Gait & posture*, vol. 28, no. 2, 2008, pp. 285-291.

[29] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” *In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, August 2016, pp. 785-794.

[30] K. Han, Q. Yang, and Z. Huang, “A two-stage fall recognition algorithm based on human posture features,” *Sensors*, vol. 20, no. 23, 6966, 2020.

[31] P. Vallabh and R. Malekian, “Fall detection monitoring systems: a comprehensive review,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 6, 1809-1833, 2018.

[32] T. Huang, “機器學習: Ensemble learning 之 Bagging、Boosting 和 AdaBoost,” Medium, June 20, 2018. [Online]. Available: https://chih-sheng-huang821.medium.com/%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92-ensemble-learning%E4%B9%8Bbagging-boosting%E5%92%8Cadaboost-af031229ebc3

[33] K. Rocha, “People walking on pedestrian lane during daytime,” (n.d.). [Online]. Available: https://www.pexels.com/photo/people-walking-on-pedestrian-lane-during-daytime-109919/

[34] “Falls,” World Health Organization, April 26, 2021. [Online]. Available: . https://www.who.int/news-room/fact-sheets/detail/falls

[35] M. Antonello, M. Carraro, M. Pierobon, and E. Menegatti, “Fast and robust detection of fallen people from a mobile robot,” *In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2017, pp. 4159-4166.

[36] S. G. Miaou, P. H. Sung, and C. Y. Huang, “A customized human fall detection system using omni-camera images and personal information,” *In 1st Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare*, April 2006, pp. 39-42.

[37] R. Luque, E. Casilari, M. J. Morón, and G. Redondo, “Comparison and characterization of android-based fall detection systems,” *Sensors*, vol. 14, no. 10, 2014, pp. 18543-18574.

[38] B. Andò, S. Baglio, C. O. Lombardo, and V. Marletta, “A multisensor data-fusion approach for ADL and fall classification,” *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 9, 2016, pp. 1960-1967.

[39] M. Á. Á. De la Concepción, L.M.S. Morillo, J. A. Á, García, and L. González-Abril, "Mobile activity recognition and fall detection system for elderly people using Ameva algorithm," *Pervasive and Mobile Computing*, vol. 34, 2017, pp. 3-13.

[40] "Fall Detection System Market," *Persistence Market Research*, September 2019. [Online]. Available: https://www.persistencemarketresearch.com/market-research/fall-detection-system-market.asp

[41] R. Bhardwaj, A. R. Nambiar, and D. Dutta, "A study of machine learning in healthcare," *In 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, vol 2, July 2017, pp. 236-241.

[42] J. Silva, J. Madureira, C. Tonelo, D. Baltazar, C. Silva, A. Martins, and I. Sousa, "Comparing machine learning approaches for fall risk assessment," *In International Conference on Bio-inspired Systems and Signal Processing*, vol. 5, February 2017, pp. 223-230.

[43] "Our Aging Population: Statistics," Comfort Life, (n.d.). [Online]. Available: https://www.comfortlife.ca/retirement-communities/our-aging-population-statistics

[44] Z. Cao, G. Hidalgo, T. Simon, S. E. Wei, and Y. Sheikh, "OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields," *IEEE transactions on pattern analysis and machine intelligence*, vol 43, no. 1, 2019, pp. 172-186.

[45] A. Biswas and B. Dey, "A Literature Review of Current Vision Based Fall Detection Methods," *In International Conference on Communication, Devices and Networking*. Springer, Singapore, December 2019, pp. 411-421.

[46] Y. R. Serpa, M. B. Nogueira, P. P. M. Neto, and M. A. F. Rodrigues, "Evaluating Pose Estimation as a Solution to the Fall Detection Proble," *In 2020 IEEE 8th International*

Conference on Serious Games and Applications for Health (SeGAH), August 2020, pp. 1-7.

[47] T. Huang and H. Liu, "Acceptability of robots to assist the elderly by future designers: A case of Guangdong Ocean University industrial design students," *Sustainability*, vol. 11, no. 15, 4139, 2019.

[48] "ElliQ robot by Yves Behar helps ageing adults stay connected to the world," *YouTube*, January 11, 2017. [Online]. Available: https://www.youtube.com/watch?v=jsffC2gWCow

[49] "The Mabu Personal Healthcare Companion from Catalia Health," *YouTube*, December 21, 2017.[Online]. Available: https://www.youtube.com/watch?v=A3XwzlvOW7k

[50] "Meet a robot offering care and companionship to seniors," *YouTube*, June 7, 2018. [Online]. Available: https://www.youtube.com/watch?v=Qt98NlE_SRo&t=87s

[51] "The Shortage of Personal Support Workers in Canada – Is There A Solution," *Living Assistance Services*, February 1, 2019. [Online]. Available: . https://www.laservices.ca/the-shortage-of-personal-support-workers-in-canada-is-there-a-solution/

[52] M. Bosch-Jorge, A. J. Sánchez-Salmerón, Á. Valera, and C. Ricolfe-Viala, "Fall detection based on the gravity vector using a wide-angle camera," *Expert systems with applications*, vol 41, no. 17, 2014, pp. 7980-7986.

[53] L. Hazelhoff and J. Han, "Video-based fall detection in the home using principal component analysis," *In International Conference on Advanced Concepts for Intelligent Vision Systems*. Springer, Berlin, Heidelberg, October 2008, pp. 298-309.

[54] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*,2020.

[55] A. Núñez-Marcos, G. Azkune, and I. Arganda-Carreras, “Vision-based fall detection with convolutional neural networks,” *Wireless communications and mobile computing*, 2017.

[56] K. De Miguel, A. Brunete, M. Hernando, and E.Gambao, “Home camera-based fall detection system for the elderly,” *Sensors*, vol 17, no. 12, 2864, 2017.

[57] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, “Monocular 3D head tracking to detect falls of elderly people,” *In 2006 international conference of the IEEE engineering in medicine and biology society*, August 2006, pp. 6384-6387.

[58] M. Karbasi, S. Bilal, R. Aghababaeyan, A.E. Rad, Z. Bhatti, and A. Shah, “Analysis and Enhancement of the Denoising Depth Data Using Kinect Through Iterative Technique,” *Jurnal Teknologi*, vol. 78, no. 9, 2016.

[59] S. W. Yang and S. K. Lin, “Fall detection for multiple pedestrians using depth image processing technique,” *Computer methods and programs in biomedicine,* vol. 114, no. 2, 2014, pp. 172-182.

[60] K. P. Kamble, S. S. Sontakke, H. Donadkar, R. Poshattiwar, and A. Ananth, “Fall Alert: A Novel Approach to Detect Fall Using Base as a YOLO Object Detection,” *In International Conference on Advanced Machine Learning Technologies and Applications.* Springer, Singapore, February 2020, pp. 15-24.

[61] “Zenbo -Your Smart Little Companion,” *YouTube*, August 19, 2016. [Online]. Available: https://www.youtube.com/watch?v=lzVHDgItx6o

[62] Z. Cao, G. Hidalgo, T. Simon, S.E. Wei, and Y. Sheikh, “OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 1, 2019, pp. 172-186.

Appendices

Appendix A.

OpenPose Demo

Import library

```
import pyopenpose as op
import cv2
from matplotlib import pyplot as plt
```

Set up pose estimation class

```
class PoseEstimator:
    def __init__(self):
        # Custom Params (refer to include/openpose/flags.hpp for more parameters
        self.params = dict()
        self.params["model_folder"] = "/openpose/models/"

        # Starting OpenPose
        self.opWrapper = op.WrapperPython()
        self.opWrapper.configure(self.params)
        self.opWrapper.start()

        # Process img
        self.datum = op.Datum()

    def pose_estimation(self, img):
        self.datum.cvInputData = img
        self.opWrapper.emplaceAndPop(op.VectorDatum([self.datum]))

        return self.datum.poseKeypoints, self.datum.cvOutputData
```

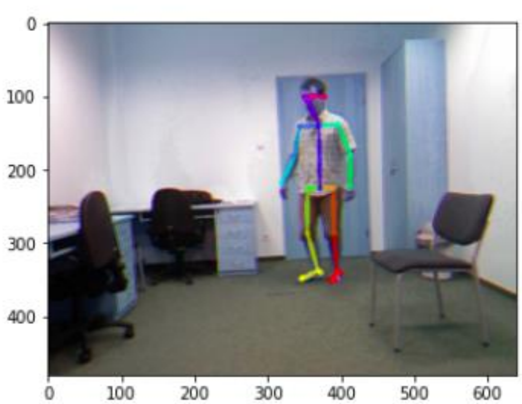
Prediction on single person image

Pose_estimation method returns two results, first is keypoints, second is image with skeletion.

```
img = cv2.imread('image/fall-01-cam0-rgb-078.png')
pose_estimator = PoseEstimator()
keypoints, output_img = pose_estimator.pose_estimation(img)
```

```
plt.imshow(output_img)
```

```
<matplotlib.image.AxesImage at 0x7ff00027feb0>
```



Single person's keypoint result
keypoints.shape = (number of people, 25keypoints, Each keypoint has 3 values)
Each keypoint's value is [x_coordinate, y_coordinate, confidence_value]

```
print(keypoints.shape)
print(keypoints)
```

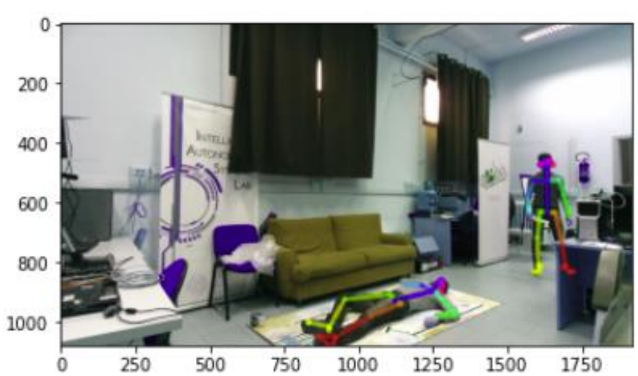
```
(1, 25, 3)
[[[355.504      107.55014      0.8692669 ]
  [369.91583    140.13855      0.93139225]
  [343.78046    141.4162       0.85200125]
  [335.9171     183.17574      0.8675577 ]
  [319.02954    217.18884      0.822407  ]
  [395.94708    138.85341      0.81443846]
  [412.91714    179.33208      0.85504824]
  [412.90277    222.40091      0.8853183 ]
  [371.15408    224.97748      0.80608976]
  [354.19507    225.00172      0.74147105]
  [354.21756    286.27957      0.83318233]
  [368.571      337.21207      0.8667247 ]
  [389.42517    224.95798      0.7585034 ]
  [390.76404    286.27478      0.90621877]
  [394.651      338.5304       0.85612947]
  [350.28256    100.98082      0.9143667 ]
  [360.7059      99.68992      0.9477244 ]
  [  0.           0.           0.        ]
  [377.69357    101.01104      0.8391236 ]
  [386.85077    350.2763       0.8437903 ]
  [393.39865    350.2346       0.8212042 ]
  [399.8471     341.19742      0.745439  ]
  [346.39053    348.9824       0.8386359 ]
  [347.67874    346.33694      0.7440074 ]
  [372.47598    339.90186      0.8812938 ]]]
```

Prediction on two people image

```
img = cv2.imread('image/test/image_0061.png')
pose_estimator = PoseEstimator()
keypoints, output_img = pose_estimator.pose_estimation(img)
```

```
plt.imshow(output_img)
```

```
<matplotlib.image.AxesImage at 0x7fef241d4580>
```



Two people's keypoint result

```
print(keypoints.shape)
print(keypoints)
```

```
(2, 25, 3)
[[[1.6369836e+03 4.7147278e+02 9.0639246e-01]
  [1.6339377e+03 5.2718689e+02 9.5575833e-01]
  [1.5898510e+03 5.2722107e+02 9.4252348e-01]
  [1.5692714e+03 5.8039203e+02 8.9025259e-01]
  [1.5721636e+03 6.2741119e+02 8.7184399e-01]
  [1.6752191e+03 5.2434943e+02 8.6660409e-01]
  [1.6899188e+03 5.7743024e+02 9.0970230e-01]
  [1.7074877e+03 6.3916302e+02 9.0793157e-01]
  [1.6340428e+03 6.3329565e+02 8.4953082e-01]
  [1.6102710e+03 6.3042529e+02 8.0474472e-01]
  [1.6015674e+03 7.1868530e+02 8.3281469e-01]
  [1.6046244e+03 8.0979077e+02 8.8139802e-01]
  [1.6604912e+03 6.3325806e+02 8.2701135e-01]
  [1.6811171e+03 7.1861609e+02 8.9313358e-01]
  [1.6958491e+03 8.0984564e+02 6.4410877e-01]
  [1.6310814e+03 4.6253839e+02 9.2776114e-01]
  [1.6429075e+03 4.5979813e+02 9.0795499e-01]
  [1.6105150e+03 4.7430835e+02 9.3522978e-01]
  [1.6546018e+03 4.7435263e+02 2.8322053e-01]
  [1.7076333e+03 8.3349414e+02 3.6657429e-01]
  [1.7223402e+03 8.3048016e+02 4.0196601e-01]
  [1.6870273e+03 8.1872424e+02 4.7722995e-01]
  [1.6074868e+03 8.3342542e+02 9.3617344e-01]
  [1.5898167e+03 8.3051709e+02 9.3611169e-01]
  [1.6103962e+03 8.1285193e+02 8.3921939e-01]]
[[1.2749185e+03 8.6284796e+02 7.6672018e-01]
 [1.2455491e+03 8.9821613e+02 7.4718052e-01]
 [1.2132333e+03 8.7757574e+02 6.8778867e-01]
 [1.1601417e+03 8.7466217e+02 1.0733417e-01]
 [0.0000000e+00 0.0000000e+00 0.0000000e+00]
 [1.2749481e+03 9.1876434e+02 7.5462586e-01]
 [1.3279589e+03 9.7467517e+02 9.1834629e-01]
 [1.2777444e+03 9.8065350e+02 8.5853887e-01]
 [1.1248489e+03 9.2750842e+02 6.8713188e-01]
 [1.1072029e+03 9.0705817e+02 6.3232011e-01]
 [9.5410944e+02 9.2458527e+02 8.4315753e-01]
 [9.0393799e+02 1.0158921e+03 6.6276640e-01]
 [1.1424882e+03 9.3936359e+02 6.3319677e-01]
 [1.0423821e+03 9.8056061e+02 7.9345024e-01]
 [9.2458789e+02 1.0513845e+03 6.2990820e-01]
 [1.2807228e+03 8.5987793e+02 4.9648416e-01]
 [1.2837086e+03 8.7461206e+02 6.9746602e-01]
 [0.0000000e+00 0.0000000e+00 0.0000000e+00]
 [1.2838804e+03 9.0103784e+02 6.6677034e-01]
 [8.6289905e+02 1.0452155e+03 2.2346225e-01]
 [8.9817981e+02 1.0748533e+03 2.2912990e-01]
 [9.1582678e+02 1.0748438e+03 3.7790585e-01]
 [8.3335400e+02 1.0013077e+03 6.4194721e-01]
 [8.3352124e+02 1.0011056e+03 6.3040644e-01]
 [9.0104431e+02 1.0276755e+03 2.7859685e-01]]]
```

Appendix B.

Scikit-learn Demo

Divide dataset into Training and Testing

```
# Divide dataset into Training and Test
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2)
```

Perform oversampling method on training data

We use SMOTE Tomek as our oversampling method in this example

```
# SMOTE Tomek
from imblearn.combine import SMOTETomek
smt = SMOTETomek(random_state=42)
X_res, y_res = smt.fit_resample(X_train, y_train)
```

Feature Scaling

```
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Train Machine Learning model

We train KNN model in this example.

```
# Train KNN model
from sklearn.neighbors import KNeighborsClassifier

knn_model = KNeighborsClassifier(n_neighbors=3, weights='distance', p=1)
knn_model.fit(X_train, y_train)

y_predict = knn_model.predict(X_test)
```

Evaluation on Imbalanced dataset.

```
from imblearn.metrics import classification_report_imbalanced, confusion_matrix

# confusion matrix
cm = confusion_matrix(y_test, y_predict)
print(cm)

# imbalanced data report
from imblearn.metrics import classification_report_imbalanced
classes = ['Normal', 'Fall', 'Lying']
print(classification_report_imbalanced(y_test, y_predict, target_names=classes))
```

```
[[53958    96    94]
 [  335   966   278]
 [  189    79  3854]]
                  pre       rec       spe        f1       geo       iba       sup

     Normal       0.99      1.00      0.91      0.99      0.95      0.91     54148
       Fall       0.85      0.61      1.00      0.71      0.78      0.59      1579
      Lying       0.91      0.93      0.99      0.92      0.96      0.92      4122

avg / total       0.98      0.98      0.92      0.98      0.95      0.91     59849
```