CS Thesis Title

A Thesis

presented to

School of Applied Computing, Faculty of Applied Science and Technology

of

Sheridan College, Institute of Technology and Advanced Learning

by

Jane R. Smith

in partial fulfilment of the requirements

for the degree of

Honours Bachelor of Computer Science (Mobile Computing)

June 2020

CS Thesis Title

by

Jane R. Smith

Submitted to the School of Applied Computing, Faculty of Applied Science and Technology
on December 13, 2020, in partial fulfillment of the
requirements for the degree of
Honours Bachelor of Computer Science (Mobile Computing)

Abstract

Around the world, falls among the elderly are occurring more frequently than during any other time in history. Unlike other age groups, falls in the elderly pose serious health risks due to the increased potential for health-related consequences that transpire after a fall has occurred. Fall-related injuries currently cost $34 billion direct healthcare costs in the USA. In Canada and other countries, direct medical expenses are in the tens of billions. Fall alarm and fall detection systems exist but have limitations in terms of accuracy and efficacy. In this paper we describe an innovative fall detection algorithm and report on real test case scenarios. Our solution offers the following benefits: 1) it is a lightweight application that can run on any low-powered device such as Arduinos, or Raspberry Pis; 2) it unobtrusively monitors the senior and requires no involvement from the resident; 3) it is easily deployed to any home or senior living environment; 4) it does not require the senior to wear a special wearable; and 5) it is secure and confidential—all processing and decision making is done locally—only notifications indicating that a fall has been detected are transmitted to stakeholders. In real test case scenarios, our system was able to achieve 97% accuracy compared to ground truth.

Keywords: fall detection, falls in the elderly, human pose estimation, Smart Healthcare, machine learning, computer vision, seniors.

Thesis Supervisor: Dr. FirstName LastName
Title: Professor, School of Applied Computing

Acknowledgments

This is the acknowledgements section. You should replace this with your own acknowledgements.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Elderly, 65 years and older, represent the fastest growing segment of the population worldwide [1]. In the United States, this was 13% in 2010 and is expected to reach 20.2% by 2050. In Europe, this was 22.5% in 2005 and is expected to reach 30% by 2050 [2]. Worldwide, the population of elderly individuals over 80 years old is over 140 million and is expected to more than triple that by 2050, increasing to nearly half billion [1]. The World Health Organization reports that nearly 1 out of 3 seniors will have a fall incident each year [3]. Around the world, falling incidences involving the elderly are quite common and detrimental to the health of those that have fallen [4]. Fall-related injuries currently cost the Canadian health care system $2.8 billion per year [5]. Even a 20% reduction in the rate of falls among Canadian seniors would translate into approximately 7,500 fewer hospitalizations and 1,800 fewer permanently disabled seniors, for an overall national savings of as much as $138 billion per year [5].

The first fall by a senior is usually the first fall of many to come, due to the fear of falling again and the gradually weakening bones of older human bodies [6]. These weaker bones make harsher injuries after a fall more likely, especially when much of the time when people fall – they are left alone without help for a period of time. It could take hours, or even a day, before someone discovers that this person has fallen when they live alone at home as many elderly citizens do [6]. Known as a *long lie*, these long periods of laying on the floor waiting for support are likely to be terrible for health, pointing to more and more severe falls in the future [4, 7].

In this paper, we address the problem of elderly falling and the *long lie* by describing a computer vision machine learning (ML) and human pose analysis fall detection and alert system. The system is used to detect a fall happening in the area, and can send detection notifications securely to family members, healthcare providers, and/or appropriate caregiver, immediately when the fall happens.

In this work, we present our fall detection system, specifically designed to be a lightweight application, unobtrusive, and easily deployed to many home or senior living facility environments. Furthermore, our system promotes senior independence as there is no reliance on the elder to use wearables whatsoever. While having devices on the person would likely increase the accuracy of a fall detection system, it does not help covering cases where the elderly might forget to wear a device, refuse to wear it, or not charge it before walking around their homes [8]. The fall detection system presented in this paper meets these desirable characteristics through the use of camera footage, computer vision and machine learning. To our knowledge, there are no such systems that provide these features; this provides motivation and relevance for this research study. We validated our system against these desirable characteristics:

1. The solution must be a lightweight application requiring low computational resources;

2. Unobtrusively monitors the senior;

3. Easily deployed to any home or senior living environment;

4. No reliance on the resident to wear a device or wearable; and

5. Must be secure and confidential.

1.1 Thesis Statement

This is where you would state your thesis. A thesis statement focuses your ideas into one or two sentences. It is crisp and to the point. A thesis statement describes the scope, purpose, and direction of the paper. It summarizes the conclusions that you have reached about the topic.

1.2 Outcomes and Contributions

We addressed the desirable characteristics, designed, implemented and tested our system with young participants. The main contributions of this work are as follows. Our system

1. is extremely lightweight; it can be run in a browser on any tablet, smartphone, or any low-powered device such as an Arduino, or Raspberry Pi;
2. discreetly monitors the senior in his/her home and requires no involvement from the senior;
3. can be easily deployed to any home or senior living environment;
4. has no reliance on the senior to wear a wearable in contrast to other systems that require a bracelet or pendant with a push button alarm to be worn;
5. is secure and confidential. All processing and decision making is done locally—only notifications indicating that a fall has been detected is transmitted to stakeholders; and
6. is accurate at detecting falls – 97% accuracy across a diverse set of fall scenarios involving 9 participants.

Chapter 2

Literature Review

Given the aging population, it is no surprise that falls among seniors are occurring at an increasing rate [3, 2, 9]. Recognized as a global phenomenon, aging demographics for people 65+ is growing steadily and so is the susceptibility—and consequences of falls [3, 2, 9]. According to the World Health Organization, nearly 1 in 3 people over the age of 65 falls each year, resulting in 29 million reported falls and admits to medical clinics and hospitals in the USA alone [3]. Unlike other age groups, falls in the elderly pose serious health risks due to the increased potential for health-related consequences that transpire after a fall has occurred. Fall-related injuries currently cost $34 billion (USA), $14 billion (Canada), with comparable expenditures for other countries [2]. Despite such a significant problem, there are few implemented solutions that address this global issue. This review explores general literature review for Falls for Seniors; and Fall Detection Methods.

2.1 General Literature Review of Seniors Falling

The main factors that place elderly individuals at increased risk for falling are: 1) biological, 2) behavioural, 3) environmental, and 4) socio-economic [5]. These groupings are not mutually independent as several risk factors may be applicable to more than one category. Biological risk factors include advanced age, gender, chronic and acute health conditions, cognitive impairments, lower-extremity weakness, physical limitations, gait abnormalities, balance deficits, and altered sensation [5]. Behavioural factors include a history of previous falls and the use of medications especially for persons using four or more prescription drugs [5]. Additional behavioural risk factors include the use of alcohol and other non-medical drugs, fear of falling, poor diet, insufficient exercise, inappropriate footwear, and the use of assistive devices [7]. Environmental risk factors are extremely varied and include poor weather conditions, uneven sidewalk surfaces, slippery floors, cluttered furniture, poor lighting, and use of unsafe equipment such as wheeled beds or chairs [10]. Socio-economic factors that increase the risk of falling include inadequate housing, low income, lack of social support, and social isolation [6]. Since falls sustained by elderly are generally caused by a combination of risk factors, it is important that health care practitioners consider all four categories when developing management strategies to prevent falls from occurring [5, 7, 10].

2.2 Fall Detection Methods

Fall-detection methods are roughly categorized into four groups: wearable/mobile-device based, radio-wave based, pressure-sensor based, and vision based [11].

2.2.1 Wearable/mobile-device based

There has been much research into the topic of fall detection alert systems and many different types of implementations exist. Most of the time, wearables are used for detection, especially an accelerometer, sometimes combined with a gyroscope [4, 8, 12]. Virtually all modern smartphones have accelerometers built into them with high accuracy, sensitivity and specificity [13]. Thresholds have been identified based on empirical studies from the accelerometer and gyroscope fall data, so a fall will be triggered when specific conditions are met [13]. The drawbacks of these systems are that they require the senior to have the device on his/her person and have it charged up. If the device is not charged or the senior forgets to carry it, the fall detection system will not function. For instance, the Apple Watch now offers a Fall Detection service. This wearable provides high sensitivity and specificity, but it also has some drawbacks. Seniors may forget to wear their Apple Watch or forget to recharge it. According to Apple, the Watch should be recharged every 18 hours [2]. Studies have shown this is a severe limitation to any practical implementation [2].

2.2.2 Radio-wave based

Radio frequency signals have also been used to detect human poses, some using reflection/refraction and changes to the signal to infer location and pose [14]. Others use RF tags placed at key points on a body to determine if a person has fallen [15]. Finally, like the system described in this paper, there are implementations that use cameras and computer vision to analyze footage and try to pinpoint key locations on the body, either frame by frame, or over a series of frames, or through a keypoint heatmap, to determine if poses by people in the footage seem to imply that a fall has occurred [16]. Besides a large number of threshold-based systems, most of these implementations make use of machine learning to teach the system what a fall might look like [16]. Radio-wave based human pose analysis is quite new and no fall detection systems have been implemented that use this approach [17]. The main limitation of this approach is that it is entirely based on the WiFi frequency in the living environment. Consequently, substantial on-site training is needed for the algorithm which may be prohibitive for wide-scale implementation in senior living environments and senior's homes.

2.2.3 Pressure-sensor based

Pressure-sensor based fall detection systems detect the impact of the body with the ground or the near horizontal orientation of the faller following a fall [4]. Most fall-detection systems detect the shock received by the body upon impact using accelerometers [18]. For example, Diaz et al. [13] developed a primary fall-detection system that involved a small adhesive sensor attached to the sacrum (the shield-shaped bony structure located at the base of the lumbar vertebrae). Its fall detection accuracy was

100% with only 7.5% of activities of daily living (ADL) predicted incorrectly as falls. Hwang et al. used a tri-axial accelerometer and gyroscope, both positioned on the chest, to differentiate between falls and ADL [19]. This system had a sensitivity of 95.5% and specificity of 100%. However, ADL testing of the system was only for three young adults who performed sitting and a daily life activity. To date fall-detection systems have used young participants to test the extent of misdetection of ADL as falls. Elderly people often move differently than younger people as they typically have less control over the speed of their body movements due to reduced muscle strength with old age.

Unfortunately, all pressure-sensor based systems have the same practical limitations. They require the senior to wear a patch or device that needs to be charged on a frequent basis (typically daily), is prone to damage and is unlikely to be worn on a consistent and continual basis [4, 11, 18, 19].

2.2.4 Vision based

Recent advancements in computer vision, machine learning, and deep learning, have provided a path forward to new and more accurate fall detection vision-based systems. Such systems are showing considerable success [20, 21]. Machine Learning vision-based approaches for fall detection have been implemented by using training sets of different states a person can be in; namely, *standing, walking, sitting, crouching, lying down* and other activities of daily living and states directly related to falling, namely, *falling*, and *fallen* [12, 14, 15]. Fall detection machine learning algorithms have been implemented using Support Vector Machines (SVMs) [22, 23, 24], kNN [25], and Naive Bayes [26]. The reported accuracy and sensitivity of these systems ranged from 82% to 95%.

However, like the other methods, there are some limitations in terms of practicality [27]. For instance, as most falls tend to occur in the bathroom [1], it may not be appropriate to have video camera in this location [27].

2.2.5 Human Pose Estimation

Human pose estimation is defined as the problem of determining the location of human joints (keypoints), such as knees, elbows, wrists, etc. in images or videos [28, 29]. It is also defined as the search for specific poses in the search space of all articulated poses [28, 29]. Human pose estimation is a new method attracting attention in the area of fall detection. When using specific cameras and computer vision for a fall detection system, pose data of a person can be used to train a fall detection ML system. Currently, there are two prominent pose detection algorithms available for researchers to create fall detection systems, OpenPose [28] and PoseNet [30]. Fall detection systems have been created using OpenPose and have reported accuracies in the range of 85%-96% [31]. However, the computational requirement to run the detectors with OpenPose is substantial [28]. PoseNet is a much lighter weight algorithm that requires significantly less computational resources [30]. It offers a comparable degree of accuracy at pose estimation; however, it can be run on simple low-powered devices (e.g., Arduino for instance). This makes it attractive for practical implementations in the home or senior living environments. At this time, there are no reported studies that have created a fall detection system using PoseNet. These considerations make PoseNet a promising approach worthy of consideration for this problem.

Chapter 3

Methodology

This methodology is divided into two sections: the method by which we simulated falls to acquire data to train our algorithm, and the method by which we created and evaluated our fall detection system.

3.1 Participants

Nine volunteers were involved in this experiment (all males). Participants had a median age of 21 (mean 26.5, min. 19, max. 49). All participants were sampled from the general population of an educational institution in the Greater Toronto Area in Ontario, Canada. Participants were recruited by a set of posters posted at various locations throughout the campus. The recruitment message did not disclose the purpose of the experiment but described the task as fun. The message indicated that each volunteer would receive compensation for his/her time and that volunteering would be a contribution to the advancement of science.

3.2 Simulated Fall Study

Unlike many areas of Machine Learning, where there is currently a rapidly growing number of datasets for scientists to explore deep learning and conduct research in machine learning, there are very few datasets for falls and none for falls of seniors. We found three public fall datasets:

1. Multiple Cameras Fall Dataset [32] contains 24 scenarios recorded with 8 video cameras from different perspectives in a room. The first 22 scenarios contain a fall and activities of daily living, the last 2 ones contain only ADL events.

2. UR Fall Detection Dataset [33] contains 30 falls and 40 activities of daily life activities. Fall events were recorded by 2 Microsoft Kinect cameras while non-fall events were recorded by only one device and an accelerometer.

3. The Fall Detection Dataset [34] is a dataset in realistic video surveillance data recorded by a single camera. This dataset consists of 191 labeled videos at 25 FPS and 320x240 resolution, representing the fall-down position in the video sequence. This dataset is characterized by its multiple scenarios, including home, coffee room, office and lecture room, which can better test the robustness of fall detection algorithms.

The *Multiple Cameras Fall Dataset* has an extreme fisheye view in all videos which distorts the images; the resolution in the last two (*UR Fall Detection* and the *Fall Detection Dataset*) is very low. Unfortunately, these limitations prohibited us from using them in our study. Furthermore, these datasets did not include any aids (i.e., canes, walkers, etc.) that a senior would typically have, especially if s/he has balance issues. Consequently, this first portion of our methodology is a simulated fall dataset collection process.

In order to train our machine learning model, we created video footage of various falling and non-falling scenarios in a safe controlled environment as done by our young participants as it would not be appropriate to subject elderly people to simulated falls. A large crash mat was used to ensure the safety of the participants as they performed the prescribed routine. Each participant performed 7 different fall types and each fall-type was repeated 2 times. Thus, each young adult performed 14 falls. All gave written informed consent and the educational institution's Research Ethics Board approved the protocol. Each participant was instructed to perform the following protocol while being video recorded:

3.2.1 Non-fall scenarios:

- Rising up from a chair (sitting position to a standing position),
- Sitting down in a chair from standing position,
- Walking around a desk in a large circle,
- Walking around in a large circle, towards and away from the camera,
- Walking around in a large circle with a cane, towards and away from the camera,
- Walking around in a large circle with a walker, towards and away from the camera.

3.2.2 Fall scenarios:

- Fall while initially leaning with a hand on a desk (e.g., simulating falling beside a counter in a kitchen for instance),
- Falling beside a desk,
- Falling from behind a desk,
- Stumbling backwards onto the ground after a perceived loss of balance,
- Falling forward from a chair after standing up,
- Falling to the side with canes, and lastly,
- Falling to the side with walkers.

More variety in video data from a diverse collection of angles has been shown to improve accuracy in machine learning models [32, 33, 34]. Hence, all data collected was taken at multiple angles for filming. As for walking in a large circle, this was to have a variety of far away and close-ups of people relative to the camera while affording multiple perspectives and gait data collection. This also was based on best practices for machine learning researchers curating their own image datasets [32, 33, 34]. The use of the equipment (i.e., a cane and a walker), was incorporated into the protocol to better simulate walking patterns, gait, balance issues and appearances of seniors who likely use such aids, especially if they are at high risk for falling. The non-fall scenarios were filmed to create a comprehensive fall dataset and enabled extensive training, validation and testing of our ML models.

A total of 235 videos were collected, with 115 being non-fall scenarios, and 120 being fall scenarios, collectively representing 36.4 minutes of footage. Each participant had on average 26 videos of their own actions, and the total data from all participants were used to train our ML models. Using conventional ML practices, 80% of the video data was used for training and 20% was for testing our models. Figure 3-1 presents a) a participant falling forward, b) a participant falling backwards onto the mat, and c) a participant using a cane and falling to the side.

3.3 Design

In this section, we describe the process we followed to build a machine learning model that can detect falls. The machine learning model that fits this need is a logistic regression model that is based on a supervised classification algorithm. As shown in Figure 3-2, our model is defined as following:

- Input features (X): the input needed for our fall detection model are the keypoint positions that identify the pose of a single person.

- Target variable (y): the output of our model is a class that predicts the state of the person in the video (based on the input features). We restricted the classes in our model to four different classes: *standing, sitting, falling* and *lying.* During the initial stages of our ML research in creating a fall detection model, we included many other classes that were being tracked too, such as *running,*

Figure 3-1: A participant falling forward (a), a participant stumbling backward (b), and a participant using a cane and falling sideways (c).

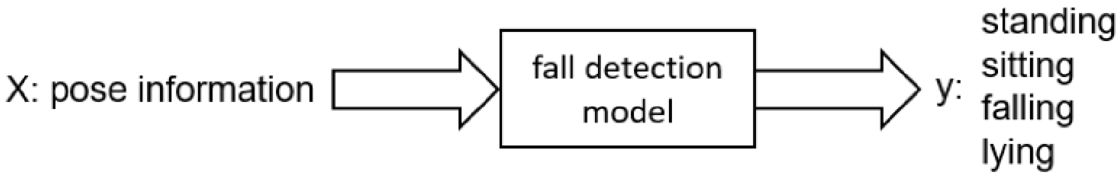

Figure 3-2: Machine Learning Model for Fall Detection.

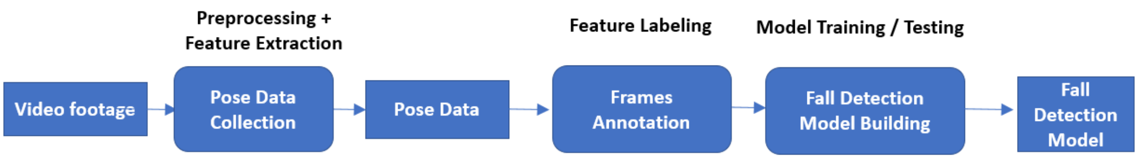

Figure 3-3: Main Pipeline Used to Build a Fall Detection Algorithm Using ML and Human Pose Estimation.

stretching, crouching, etc. but these were removed from our output class to make the detection problem simpler. Instead, we used two classes only (*Falling, Not Falling*), and discovered that having a limited amount of variation would help our history model (as described in section 3.3.3) to learn when determining a fall state.

Figure 3-3 shows the overall process we followed to build our fall detection model using ML and human pose estimation. We describe the main steps and the choices we made in the following sections.

3.3.1 Pose Data Collection

In order to get pose information from a person in a video, we made use of two open-source libraries that can be used to estimate the pose of a person in an image or a video by estimating where key body joints are. The two libraries we used are PoseNet [30] and OpenPose [28]. PoseNet is developed under MIT licence. It provides a lightweight implementation that uses TensorFlow JavaScript that runs in a web browser [30]. With this implementation, PoseNet is able to detect 17 pose keypoints in a single image. OpenPose is a more accurate pose detection package that is developed under an academic and non-profit, non-commercial use only. It is written in C++ using

OpenCV and Caffe. It enables the detection of up to 135 keypoints for one person in a single image.

We decided that the fall detection system would first require a data collection script to retrieve the pose data from a video regardless of using PoseNet or OpenPose. This pose data provided the input features to build the model. We wrote two different data collector implementations for this process (one for each library). The data, which represents keypoint information about positions of specific body parts on the people in the videos, was saved in JSON format for the OpenPose system and CSV format for the PoseNet system.

3.3.2 Frames Annotation

In order to train and build the models for each of the fall detection systems, we created ground truth datasets by tagging each frame of training videos fed through the data collector as a specific pose. We labeled every frame that contains pose data to one of our pre-defined classes: *standing, sitting, lying down, or falling.* These annotated poses were then compiled into a list of frame-by-frame pose tags, which, when reaching a certain threshold, was considered to be a full list. For the OpenPose fall system, the threshold was 30 frames, and for PoseNet it was 25 frames. These numbers were determined over the course of evaluating what frame count produces the most accurate fall detection results.

3.3.3 Fall Detection ML Model Building

In this process, we used the labeled pose data we collected to build the fall detection machine learning model. Our model builder generates two different models: the first of which is based on frame-by-frame analysis and the second uses a novel frame history analysis technique. These two models are shown in Figure 3-4.

The first model allows prediction of one frame's state (e.g. class), such as whether someone is *standing, falling, sitting, or lying down* in a frame of video. A history analysis model allows determination of an estimated frame state for a series of frames.

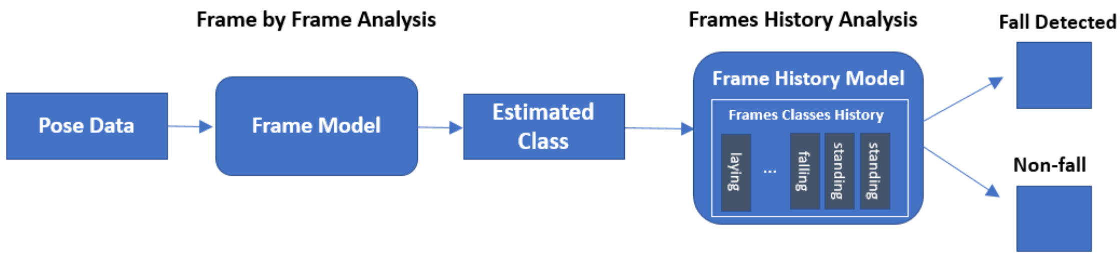

Figure 3-4: Fall Detection Model Building Process.

This second model takes in the pose estimates from the frame-by-frame prediction model's output, in a list of 25 or 30 frame state predictions, depending on the fall system being used (OpenPose or PoseNet). By looking at the past *'frame state history'* in the last few seconds of the video that was already analyzed by the first machine learning model, the second model is able to make an estimation on what state a given person is in considering past information rather than simply what is visually observable in one specific frame in the video. We explored this approach because during testing, it became clear that fall states could not be determined from one frame of a video alone – to accurately predict a real fall, one needs to know how fast the person is moving from frame to frame, and in what way. The second model we created is based on *frame history analysis* in order to address this problem.

Our novel frame history analysis technique equips the system to consider the person's position in the past as it relates to the current frame being analyzed. The frame history model's output is used to determine the final test results for both fall detection systems.

The PoseNet based model was made using *nodesvm* [28], while the OpenPose based model was made with *scikit-learn* [30]. Many OpenPose models were created through experimentation using scikit-learn's algorithms, namely, Linear SVC, Naïve Bayes, SVC, and Kernel Approximation (see Figure 3-5). Ultimately, the results of the experimentation were used to compare our best OpenPose model with the best PoseNet model.

The fall detection model building process generated model files that classify pose data into *fall* or *no-fall* categories. We then created a predictor script that uses these

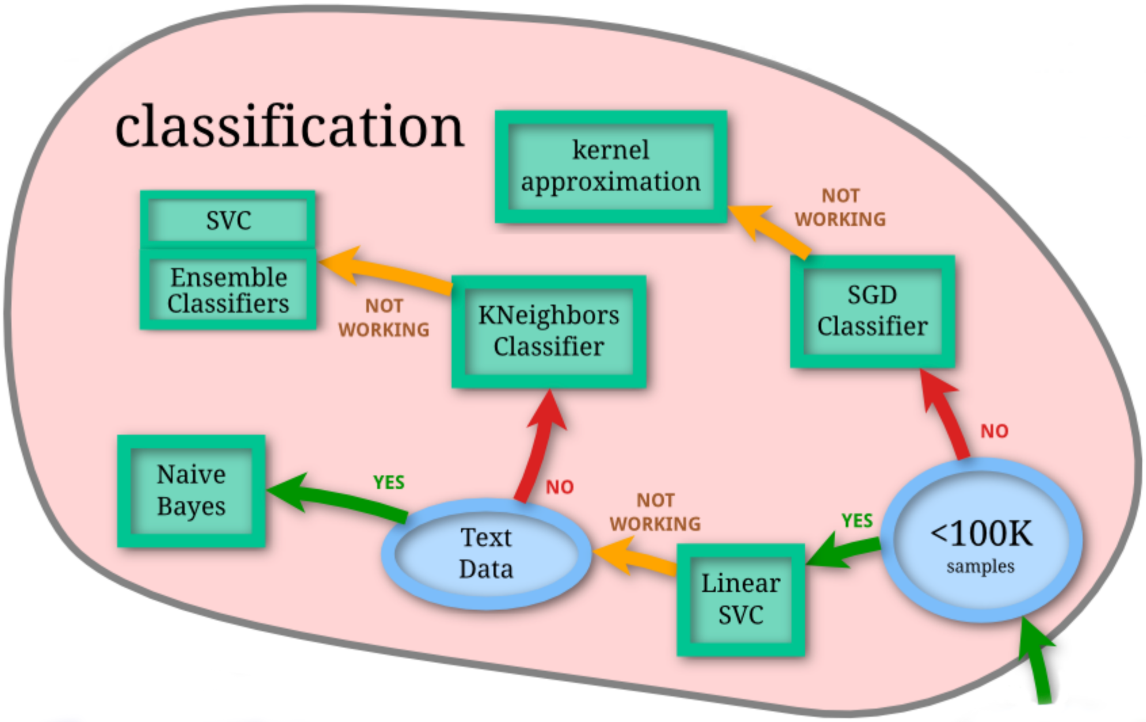

Figure 3-5: Scikit-learn Classification Road-Map.

models to make predictions about video pose data, provided by PoseNet or OpenPose, to ascertain about whether a fall is detected in a given video footage.

In order to generate statistics on the performance of both fall detection systems using either PoseNet or OpenPose, models were built for each system based on 193 videos of the total 235 videos (80% of all videos were used as training videos (randomly shuffled). 20% of the remaining videos were used for testing. These models were used with the predictor script on each system to produce predictions based on data gathered through the data collector scripts on each fall system. Videos were placed into different datasets based on which person was featured in the videos, so there were 9 groups of videos based on the 9 participants. Once prediction data were gathered, the results were organized into confusion matrices where true positives represent a true fall detection (i.e., ground truth – the frame was tagged with the person *falling*).

3.4 Analysis

The following types of quantitative analysis were performed: 1) evaluating and refining the classifier using standard machine learning evaluation tools; and 2) determining the accuracy of the classifier using statistical tools.

1. Computing confusion matrices: Compare the classifier's accuracy at predicting a fall using actual video analysis as ideal. Confusion matrices and the associated measures (Accuracy, Precision and Sensitivity) are commonly used in the evaluation of machine learning algorithms, please see: [35, 36, 37, 38, 39]. Statistics for each participant for each classifier model was collected on:

   True Positives (TP): *(classifier correctly detected a fall);*

   True Negatives (TN): *(classifier correctly selected "Not a fall");*

   False Negatives (FN): *("Didn't identify a fall when it should have");*

   False Positives (FP): *("False Alarm"—classifier said it was a fall when it shouldn't have);*

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

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

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

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

   $$FallPositiveRate(FPR) = \frac{FP}{FP + TN}$$

   $$FScore = \frac{2 * TP}{2 * TP + FP + FN}$$

2. Calculating standard descriptive statistics on the confusion matrix data for the entire group including minimum, maximum, mean, median, standard deviation, accuracy, precision, and sensitivity.

Chapter 4

Findings (Analysis and Evaluation)

4.1 Statistical Analysis to Test the Classifiers

This section presents the statistical analysis that was performed to test the effectiveness of the classifier. Numerous confusion matrix computations were performed including supporting statistical measures and summative standard descriptive statistics. The confusion matrices show the range of values (min, max, mean, median and standard deviation) across all participants. These computational results are shown in Table 1 and Table 2 based on models created from training data sets.

For our implementation using PoseNet, as shown in Table 1, on average 5.096% of results were true positives, 92.069% true negatives, 2.723% false negatives, and 0.113% false positives. The system exhibited a mean accuracy of 97.164%, precision of 98.386%, sensitivity of 65.847%, specificity of 99.877%, error rate of 2.836%, false positive rate of 0.123%, and a F-score of 78.774%.

On our other fall detection system using OpenPose, Table 2 shows, on average 0.701% of results were true positives, 96.737% true negatives, 2.519% false negatives, and 0.042% false positives. The system exhibited a mean accuracy of 97.439%, precision of 93.957%, sensitivity of 24.768%, specificity of 99.957%, error rate of 2.562%, false positive rate of 0.044%, and a F-score of 38.464%.

Table 4.1: Confusion matrix results with supporting statistical measures and summative standard descriptive statistics for PoseNet fall detection system training data set.

	True Positive (%)	True Negative (%)	False Negative (%)	False Positive (%)	Accuracy	Precision	Sensitivity	Specificity	Error Rate	FPR	F-Score
Min	2.993%	84.987%	1.503%	0.000%	94.111%	91.667%	61.429%	99.448%	1.503%	0.000%	75.605%
Max	9.125%	95.179%	5.729%	0.509%	98.497%	100.000%	73.239%	100.000%	5.889%	0.552%	84.553%
Mean	5.096%	92.069%	2.723%	0.113%	97.164%	98.386%	65.847%	99.877%	2.836%	0.123%	78.774%
Median	4.714%	92.598%	2.214%	0.042%	97.531%	99.920%	64.403%	99.957%	2.469%	0.044%	78.345%
Std Dev	1.912%	3.263%	1.377%	0.174%	1.386%	2.900%	4.485%	0.190%	1.386%	0.190%	2.856%

Table 4.2: Confusion matrix results with supporting statistical measures and summative standard descriptive statistics for OpenPose fall detection system training data set.

	True Positive (%)	True Negative (%)	False Negative (%)	False Positive (%)	Accuracy	Precision	Sensitivity	Specificity	Error Rate	FPR	F-Score
Min	0.542%	93.345%	1.080%	0.000%	93.887%	75.309%	8.233%	99.807%	1.136%	0.000%	15.060%
Max	1.049%	98.313%	6.040%	0.188%	98.864%	100.000%	39.396%	100.000%	6.113%	0.193%	56.524%
Mean	0.701%	96.737%	2.519%	0.042%	97.439%	93.957%	24.768%	99.957%	2.562%	0.044%	38.464%
Median	0.686%	97.151%	2.133%	0.010%	97.773%	98.718%	23.493%	99.990%	2.227%	0.010%	38.046%
Std Dev	0.168%	1.463%	1.506%	0.065%	1.517%	8.862%	9.291%	0.067%	1.517%	0.067%	12.249%

4.2 Summary of Main Findings

Figure 4-1 shows a chart that compares the accuracy and the sensitivity of the two fall detection models. While both fall detection systems had a generally high accuracy at determining when someone wasn't falling, as can be seen with the high true negative result and the high accuracy values of 97% on both systems, along with the high specificity values of 99%. As evidenced by the low sensitivity values of approximately 66% on the PoseNet-based fall system and 25% on the OpenPose-based fall system, actual fall detections will occur when they are supposed to, much less than 90% of the time. The low standard deviation values across most variables also implies that these percentage values are relatively constant across the variety of datasets tested on, based on the nine participants in our experiment. While the accuracy was comparable in both models, the sensitivity was significantly less in the OpenPose implementation.

Figure 4-2 presents PoseNet and OpenPose keypoints superimposed over a frame of sample video.

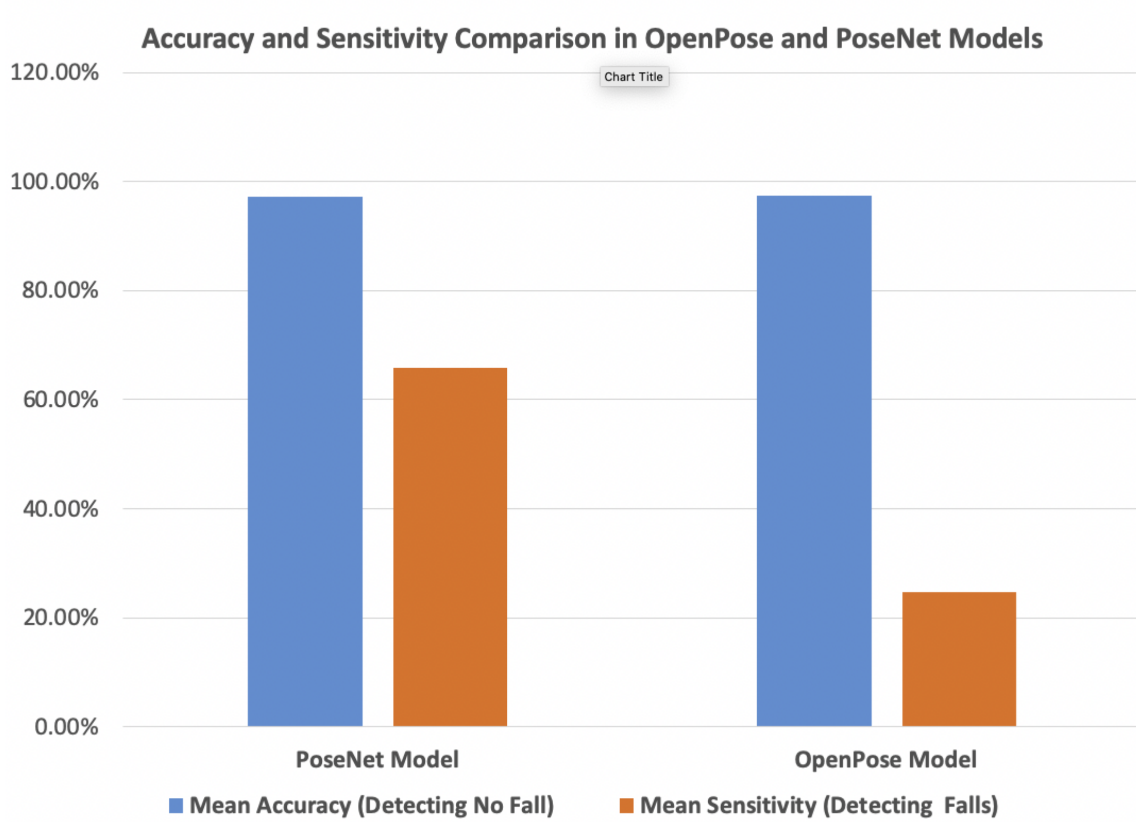

Figure 4-1: Mean Accuracy and Sensitivity for PostNet and OpenPose Based Models.

Figure 4-2: PoseNet Pose Detection over Video (top) OpenPose Pose Detection over Video (bottom).

Chapter 5

Discussion

This section presents a summary of the significant factors in designing good models, outlining the advantages and disadvantages; implications of the empirical results for deploying fall detection systems in practice; and limitations and future research.

5.1 Significant Factors in Designing Good Models

In this research, we discovered the following pros and cons with PoseNet and OpenPose for creating fall detection algorithms. During our research in creating, testing and comparing our two pose detection systems, we found that each have some drawbacks as well as advantages.

Advantages:

- PoseNet can be run on mobile devices, low computation power required
- PoseNet is free to use commercially
- OpenPose has better tracking of people partially obstructed by an object

Disadvantages:

- Both OpenPose and PoseNet have issues with detecting people where there are none, specifically on black objects like the legs of an office chair, etc.

- People falling behind objects is difficult for both OpenPose and PoseNet, though less so for OpenPose. PoseNet will lose tracking entirely with any obstruction, whereas OpenPose is able to better maintain tracking on partially obstructed people. This is likely due to the number of respective keypoints each system uses for pose estimation.

- OpenPose requires high computational power, generally runs on a powerful computer.

- OpenPose requires a substantial amount of money for commercial deployment

PoseNet ultimately led to the better fall detection results over the OpenPose based system, even though OpenPose's pose detection is slightly more accurate and richer in data (i.e., number of keypoints). It is not exactly clear why this was the case, but it is hypothesized that when people are falling, and due to their higher acceleration, pose data is more unique for OpenPose to analyze. So, this may make it easier for the OpenPose based machine learning models to differentiate *fall* frames from *non-fall* frames.

Our fall detection system with PoseNet can be run easily on mobile devices as opposed to the system using OpenPose, which requires substantially more computational power. OpenPose requires a powerful computer and simply could not be run on a mobile device at this time. Furthermore, PoseNet, unlike OpenPose, is free to use for any purpose.

Both systems exhibit issues especially when trying to detect falls when a person falls behind an obstacle, like a piece of furniture. Partial obstruction of the image of the person on video frames causes PoseNet to lose tracking entirely, while OpenPose does a somewhat better job of keeping track of the person, but still has issues. Objects in the background in frames also tend to confuse both pose detectors, especially black objects such as the legs of an office chair being confused for the legs of a person (Figure 5-1 and 5-2). Reflections in mirrors and windows, if vivid enough, can also cause false pose detections with both PoseNet and OpenPose.

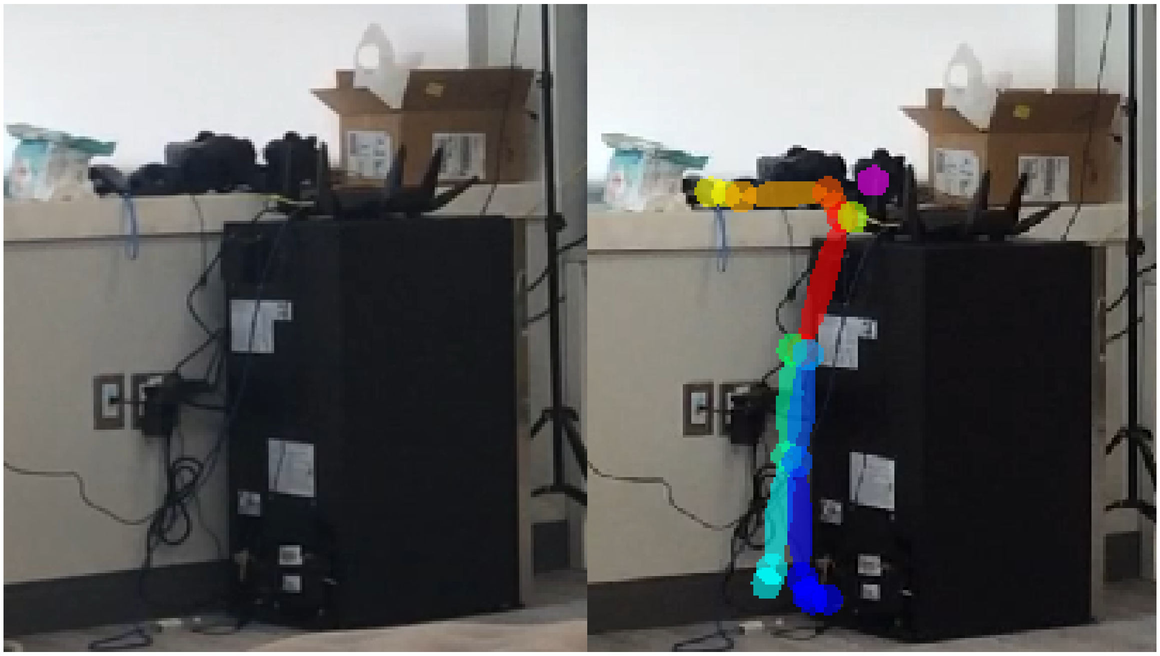

Figure 5-1: Frame from the video containing a collection of cables that strangely resemble a person (left), false identification of a person (right).

The addition of a frame history ML model analyzing the output data of the frame-by-frame data model proved to increase accuracy early on in the research that we conducted and seems to be a step in the right direction for improving fall detection based on pose analysis. The optimum size of the frame history to be analyzed has yet to be determined. While 25 and 30 frame size histories were used in our models with PoseNet and OpenPose respectively that produced good results, further experimentation should be done to determine the optimal number of frames that should be kept track of for the sake of the most accurate fall detections. Frame history was taken into consideration in this study based on the idea that a fall cannot truly be expressed in one frame in isolation from video footage. At any given frame, it is difficult to tell for sure if one is falling, or is just performing an activity of daily living (e.g., crouching, bending down, lying down, etc.). With the ability for the ML model to look through predictions based on video frames in the past, we aim to capture the moment of the person standing, starting to fall, and landing on the ground in a collapsed position.

In our experiments, we filmed participants in a variety of falling and non-falling scenarios. This brought a great deal of diversity into the dataset but was not without

Figure 5-2: Wheels and bottom structure of an office chair mistakenly identified as a person's leg.

fault. No actual elderly people like the ones this research was targeted at helping were involved in the videos filmed, and the dataset contained far too many non-falling states in comparison to the number of falling states in frames in the videos. Had the video data been carefully cut down to show only the moment of falling from start to finish, with no extra non-falling data of any kind, perhaps the results would have shown a more balanced distribution of fall state and non-fall states. Furthermore, our dataset is arguably small – more videos of moments of people falling may improve the training processes. It is not clear whether it would help, but introducing more scenarios of activities of daily living may increase the accuracy of our models, such as having people carrying things (e.g., drinking glasses, plates), or lying down in a bed, etc.

5.2 Implications for Practice

Multiple camera feeds into our algorithm would facilitate more comprehensive coverage of the person's environment and increase the overall accuracy of our system and the confidence of detecting a fall especially in these types of situations.

5.3 Limitations

This research showed that there is significant potential to create classifiers using machine learning to determine if a person has fallen or not. As with any research there are limitations. Some of the main limitations of this work are presented below.

5.3.1 Sensitivity

The lower than desired sensitivity values in both fall detection systems implies that our experimental data did not contain enough frames where participants were actually falling for the machine learning models to truly learn when falls should be detected at all times. This is evident when one compares the total true positive plus false negative portion to the total true negative plus false positive portion in Tables 1 and 2. There are significantly more non-fall frame data in comparison to fall frame data. Almost all videos predictably contained 90%+ frames where people were not in the middle of a fall, and that was likely the reason for this outcome.

5.3.2 Computational Power

The computational power required for the OpenPose based fall detection system was much higher than that needed for the PoseNet based solution. The PoseNet system has been optimized to work even on mobile devices whereas OpenPose requires a powerful system. To use a fall system based on OpenPose in practice would likely prove to be much more expensive to run than one that is based on PoseNet. This is especially a problem if one is trying to run the fall detection system in real time on a live camera feed. Running pose detection in OpenPose typically requires slowing down the video to compensate for the work the computer needs to do.

The number of participants involved in this study was sufficient for exploring how these pose detection fall prediction systems might be implemented and for discovering ways to improve upon them. However, having a much larger video dataset of say, 100 or even a 1,000 people, would likely enhance the machine learning models featured here.

5.3.3 False Detections

Even with the more sophisticated pose detection tool, OpenPose, as well as with PoseNet, there were plenty of false detections of human poses such as in background objects in the videos filmed in this research. As can be seen in Figure 5-1 for example, a fridge and a mix of cables was detected as a person by OpenPose, and in Figure 5-2, a chair next to a computer monitor was detected as another person. This would also happen with any detailed reflections in videos, if there were any on windows or mirrors, and at some points, people were falsely detected to be sitting on a couch in the background when no one was there at all. These false pose identifications in video frames throughout the datasets tested on can of course lead to worse results when fed to a machine learning model that will try to learn partially based off of the false information.

In this study, a single-pose version of PoseNet was used, and a multi-person version of OpenPose. These seemed to work best for what we were trying to study at the time, but it is possible that with further examination of other possible versions of the PoseNet/OpenPose algorithms, new results could be found. OpenPose could be configured to detect only one person in a frame, or PoseNet could be configured to detect multiple. It is likely however, that if PoseNet is configured for multiple person detection, the system would require more computational power, but it is unlikely to be as much as what OpenPose requires.

Chapter 6

Conclusion

6.1 Summary

Around the world, the elderly are falling more frequently than during any other time in history. Unlike other age groups, falls in the elderly pose serious health risks due to the increased potential for health-related consequences that transpire after a fall has occurred. Determining when authentic falls in natural home environments is an ongoing problem to which we have provided a solution. In this paper we described an innovative fall detection algorithm and report on real test case scenarios. Our solution offers the following benefits:

1. it is a lightweight application that can run on any low-powered device such as an Arduinos, or Raspberry Pi;

2. it unobtrusively monitors the senior and requires no involvement from the resident;

3. it is easily deployed to any home or senior living environment;

4. it does not require the senior to wear a special wearable;

5. it is secure and confidential—all processing and decision making are done locally—only notifications indicating that a fall has been detected is transmitted to stakeholders;

6. In real test case scenarios, our system was able to achieve 97% accuracy compared to ground truth;

7. The classifier was implemented using an advanced machine learning technology and Pose Estimation—which is a novel contribution. No other fall detection system uses human pose analysis using frame history analysis as presented in this work; and

8. This research sheds light on reasoning about fall detection in senior living environments. Currently, this is largely an unsolved problem.

6.2 Future Work

In future implementations of fall detection systems based on human pose estimation, many improvements could be made on the systems presented in this paper to aid in increasing the accuracy of fall detection. First, balancing the amount of frame data that contains fall and non-fall states from videos analyzed by cutting off the start and end points of footage to only contain the motions of falling would likely allow machine learning models to learn more appropriately what fall pose information looks like. At the same time, one would likely want to get much more video data in the future and increase the size of the training dataset used.

It may also be advantageous to investigate the effects of having a dataset that contains fully, or at least in majority, real elderly participants, to more accurately simulate falls that would happen in the targeted audience of this research. More accurate results may come about from having people who perhaps fall and walk differently from the more youthful participants in our experiments. Training datasets could also be possibly improved by including some calculations based on the pose data retrieved from the videos analyzed. For example, one could calculate the velocity of a falling participant at any given frame in a video and have this velocity value be integrated into the ML models in use to further differentiate falls from non-falls.

In the systems presented in this paper, there was often more of a problem detecting a participant's fall when they collapsed behind an object or obstacle. There are a few ways that this issue can be avoided by using multiple cameras in rooms. By viewing falls from different angles at once and confirming with each other about when a fall is truly confirmed to have happened, may make falling behind obstacles no longer a problem since most cameras would see the fall occur when one might not. Obstacles could also be treated as if they don't exist at all. For example, by using radio signals to complete pose detection through furniture, or even walls, and having these resulting poses be run through the fall detection system. In this case, there may not need to be as many cameras to confirm if a fall has occurred [17].

Finally, there is also the possibility of using other smart home devices, such as motion sensors, window/door sensors, wearable devices, and/or home voice assistants, in connected communication with each other, to further confirm that a person has truly fallen.

In the spirit of furthering science and this work, the source code for the classifier, models, and data sets will be openly available on the author's and/or journal's website. We hope this will encourage other researchers to extend and explore our work and to test and compare our classifier with other fall detection systems.

Bibliography

[1] Goodkind D. Kowal P. and He W. An aging world: 2015, international population reports. *U.S. Government Printing Office, Washington DC.*, 2016.

[2] Z. Pang, L. Zheng, J. Tian, S. Kao-Walter, E. Dubrova, and Q. Chen. Design of a terminal solution for integration of in-home health care devices and services towards the internet-of-things. *Enterprise Information Systems*, 9:86–116, 2019.

[3] World Health Organization. World report on ageing and health, 2018.

[4] Alan Bourke, J.V. O'Brien, and Gearóid ÓLaighin. Evaluation of threshold-based tri-axial accelerometer fall detection algorithm. *Gait & posture*, 26:194–9, 08 2007.

[5] Natthawadee Maneeprom, Surasak Taneepanichskul, and Alessio Panza. Falls among physically active elderly in senior housings, bangkok, thailand: situations and perceptions. *Clinical Intervention and Aging*, 13:2149–2159, 2018.

[6] Hong-Ying Pi, Meng-Meng Hu, Jie Zhang, Pei-Pei Peng, and Dan Nie. Circumstances of falls and fall-related injuries among frail elderly under home care in china. *International Journal of Nursing Sciences*, 2:237–242, 2015.

[7] Erin Miller, Elizabeth Wightman, Karla Rumbolt, Sara McConnell, Katherine Berg, Moira Devereaux, and Fiona Campbell. Management of fall-related injuries in the elderly: A retrospective chart review of patients presenting to the emergency department of a community-based teaching hospital. *Physiotherapy Canada*, 61(1), 2009.

[8] Quoc T. Huynh, Uyen D. Nguyen, Lucia B. Irazabal, Nazanin Ghassemian, and Binh Q. Tran. Optimization of an accelerometer and gyroscope-based fall detection algorithm. *Journal of Sensors*, 2015.

[9] M Horie. Change of population structure of japan and roles of administrative counselors. *National Graduate Institue for Policy Studies*, 2019.

[10] Laurence Rubenstein. Falls in older people: Epidemiology, risk factors and strategies for prevention. *Age and ageing*, 35 Suppl 2:ii37–ii41, 10 2006.

[11] Xiangbo Kong, Lehan Chen, Zhichen Wang, Yuxi Chen, Lin Meng, and Hiroyuki Tomiyama. Robust self-adaptation fall-detection system based on camera height. *Sensors*, 19(17), 2019.

[12] M.V. Albert, K. Kording, M. Herrmann, and A. Jayaraman. Fall classification by machine learning using mobile phones. *PLoS ONE*, 7(5):1–6, 2012.

[13] Jin-Shyan Lee and Hsuan-Han Tseng. Development of an enhanced threshold-based fall detection system using smartphones with built-in accelerometers. *IEEE Sensors Journal*, 19(18):8293–8302, 2019.

[14] Y. Wang, K. Wu, and Lionel Ni. Wifall: Device-free fall detection by wireless networks. 16:581–594, 02 2017.

[15] Mitja Lustrek and Bostjan Kaluza. Fall detection and activity recognition with machine learning. *Informatica (Slovenia)*, 33:197–204, 01 2009.

[16] Xiaowei Zhou, Menglong Zhu, Spyridon Leonardos, G. Derpanis, Konstantinos, and Kostas Daniilidis. Sparseness meets deepness: 3d human pose estimation from monocular video. *The Computer Vision Foundation*, pages 4966–4975, 2019.

[17] Mingmin Zhao, Tianhong Li, Alsheikh Abu, Mohammad, Tian Yonglong, Hang Zhao, Antonio Torralba, and Dina Katabi. Through-wall human pose estimation using radio signals. *The Computer Vision Foundation*, pages 7356–7365, 2018.

[18] K. Bourke, Alan, Jochen Klenk, Lars Schwickert, Kamiar Aminian, A.F. Ihlen, Espen, Sabato Mellone, L. Helbostad, Jorunn, and Lorenzo Chiari. Fall detection algorithms for real-world falls harvested from lumbar sensors in the elderly population: A machine learning approach. In *38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*.

[19] J.Y. Hwang, J.M. Kang, Y.W. Jang, and H.C. Kim. Development of novel algorithm and real-time monitoring ambulatory system using bluetooth module for fall detection in the elderly. In *26th annual international conference of the IEEE-EMBS*. IEEE.

[20] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and S. Iyengar. A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys*, 51:1–36, 09 2018.

[21] M.D. Solbach and J.K. Tsotsos. Vision-based fallen person detection for the elderly. In *IEEE International Conference on Computer Vision*, page 1433–1442. IEEE.

[22] Charalampos Doukas, Ilias Maglogiannis, Philippos Tragas, Dimitris Liapis, and Gregory Yovanof. Patient fall detection using support vector machines. volume 247, 11 2007.

[23] Homa Foroughi, Alireza Rezvanian, and Amirhossien Paziraee. Robust fall detection using human shape and multi-class support vector machine. pages 413–420, 12 2008.

[24] Shengyun Liang, Tianyue Chu, Dan Lin, Yunkun Ning, Huiqi Li, and Guoru Zhao. Pre-impact alarm system for fall detection using mems sensors and hmm-based svm classifier. volume 2018, pages 4401–4405, 07 2018.

[25] Jian He, Zihao Zhang, and Weiguo Yu. Interrupt-driven fall detection system realized via a kalman filter and knn algorithm. pages 579–584, 10 2018.

[26] G. Muneeswari, D. Daniel, and K. Natarajan. *Activity Classifier: A Novel Approach Using Naive Bayes Classification*, pages 323–330. 01 2020.

[27] C. Pramerdorfer, R. Planinc, M. Van Loock, D. Fankhauser, M. Kampel, and M. Brandstotter. Fall detection based on depth-data in practice. In *European Conference on Computer Vision*, page 195–208.

[28] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *4*, 14:1–14, 2019.

[29] Jo˜ao Carreira, Pukit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. *The Computer Vision Foundation*, 2019.

[30] Posenet: Pose estimation using posenet. https://www.tensorflow.org/lite/models /pose_estimation/ overview, 2019. [Online; Accessed 2019-12-05].

[31] Zhanyuan Huang, Yang Liu, Yajun Fang, and K. P. Horn, Berthold. Video-based fall detection for seniors with human pose estimation. In *4th International Conference on Universal Village*. IEEE.

[32] Edouard Auvinet, Caroline Rougier, Jean Meunier, Alain St-Arnaud, and Jacqueline Rousseau. Multiple cameras fall data set. 01 2011.

[33] Bogdan Kwolek and Michal Kepski. Human fall detection on embedded platform using depth maps and wireless accelerometer. *Computer Methods and Programs in Biomedicine*, 117:489–501, 10 2014.

[34] Antoine Trapet. Fall detection dataset. http://le2i.cnrs.fr/Fall-detection-Dataset?lang=fr, 2013. [Online; Accessed 2019-12-05].

[35] Charles Elkan. Evaluating classifiers, 2012.

[36] George Forman and Martin Scholz. Apples-to-apples in cross-validation studies. *ACM SIGKDD Explorations Newsletter*, 12, 11 2010.

[37] Ron Kohavi and F. Provost. Glossary of terms. special issue of applications of machine learning and the knowledge discovery process. *Mach. Learn.*, 30, 01 1998.

[38] Zhiyong lu, Duane Szafron, Russ Greiner, P Lu, D Wishart, B Poulin, John Anvik, C Macdonell, and R Eisner. Predicting subcellular localization of proteins using machine-learned classifiers. *Bioinformatics (Oxford, England)*, 20:547–56, 04 2004.

[39] H.J Hamilton. Confusion matrix. http://www2.cs.uregina.ca/ hamilton/courses/831/notes/confusion_matrix/ confusion_matrix.html/, 2011. [Online; Accessed 2019-12-05].

Appendix A

Tables

Table A.1: Armadillos

Sheridan	Mobile Computing students
are	cool!

Appendix B

Tables

Table B.1: Armadillos

Sheridan	Mobile Computing students
are	cool!