MSci Computer Science
Computer Science Department

Deep Learning Methods for the Classification of Falls within Freestyle Skiing

Candidate

Michael Hills

Student ID 700013809

Supervisor

Dr David Walker

University of Exeter

Academic Year
2023/2024

Abstract

Freestyle skiing is a winter sport that combines traditional skiing and acrobatics, resulting in a high-risk activity with a high incidence of severe injuries. Due to the risk involved and the sport's recent surge in popularity, there is a clear need for new safety measures to safeguard participants. This project aims to apply deep learning methods to classify falls and intentional movements (e.g. jumps, spins and other tricks) within the sport, thereby enabling a fast response from emergency services, enhancing the safety of freestyle skiers. The methodologies utilised are split into two main categories, image-based methods that utilise CNNs and raw sensor signal-based methods that utilise LSTM and CNN-LSTM networks. The data collection, pre-processing and deep learning neural networks utilised are outlined, along with a discussion of why each method was selected. The results for each method are subsequently provided and evaluated, addressing the advantages and limitations associated with each network along with a discussion of potential future work and improvements to the methods implemented.

	Yes	No
I certify that all material in this dissertation which is not my own work has been identified.	☑	☐
I give the permission to the Department of Computer Science of the University of Exeter to include this manuscript in the institutional repository, exclusively for academic purposes.	☑	☐

Contents

1 Introduction

1.1 Project Motivation

Freestyle skiing is a winter sport that combines traditional skiing with aerial and non-aerial acrobatics, resulting in skiers performing a variety of high-speed jumps, flips, and tricks, often in specialised terrain parks. The sport first emerged in the 1960s and, following its widespread attraction and recognition, was established as a distinct discipline by the International Ski Federation (ISF) in 1979 and was subsequently included in the Winter Olympic Games in 1988 [1]. Since then, there has been a substantial increase in the number of freestyle competitions and events and, with widespread media attention of prestigious competitions such as the Winter Olympic Games and X-Games, the sport has been propelled into the public eye. Consequently, many ski resorts have established their own terrain parks, sparking a newfound interest amongst skiers to explore and participate in freestyle skiing. [2, 3].

Freestyle skiing is, however, inherently dangerous as skiers execute intricate manoeuvres, often at high speed, with a significant risk of injury if not executed with the correct technique. Research has therefore been undertaken to retrospectively analyse the prevalence and severity of injuries with the aim of minimising harm and thus optimising the sport's enjoyment and safety. As the sport is relatively new, these studies are currently limited to world-class competitive or professional skiers. One such study revealed that among 662 world-class freestyle skiers, there was an injury rate of 44%, of which 32% were classified as severe, defined as an absence from training or competition of at least 28 days [4]. Another study, that analysed 22 publications, found an overall injury rate of 3.49 injuries per 1000 athlete-days across all skiing disciplines. However, freestyle skiing resulted in the highest incidence rate at 6.83 injuries per 1000 athlete-days, almost double that compared to traditional skiing at 3.57 [5].

With the increasing popularity of freestyle skiing and the expansion of ski resorts to include freestyle terrain parks, the sport has become accessible to less experienced skiers. Consequently, the occurrence of injuries is likely to increase. Clearly, there is a need for new safety measures and technologies to help safeguard participants, and one possible technology is the use of an automatic fall detection system. Such a system is designed to quickly identify when a skier experiences a fall and subsequently alert nearby personnel for rapid medical assistance to minimise potential consequences of serious injury. This is particularly important for those who ski independently, as it eliminates the reliance on a third-party observer to seek assistance if the skier is unable to do so due to the nature of their injury. Implementing an automatic fall detection system would therefore create a safer environment and provide peace of mind to skiers by ensuring prompt medical assistance following a significant fall.

While fall detection systems have gained significant attention in recent years, such as in cycling, running and elderly care, their application in the sport of freestyle skiing remains largely unexplored. In elderly care, these systems have proven highly effective in promptly alerting emergency services in the event of a fall, significantly reducing response times and offering peace of mind to the patient, their friends and family [6, 7]. However, integrating such a system into the dynamic and high-velocity environment of freestyle skiing presents unique challenges that necessitate further investigation. This presents an exciting research opportunity to evaluate the suitability of methods employed in current fall detection systems and their adaption for use in freestyle skiing.

1.2 Project Specification and Aims

The aim of this project is to develop a fall detection system specifically for freestyle skiing. This presents some unique challenges compared to other fall detection systems, primarily due to the frequent landing impacts and rotations experienced by skiers, that lead to false positives when using simpler, traditional methods. For example, Apple Watch's "crash detection" feature, detects a fall when the G-force experienced exceeds a certain value, that frequently generates a high number of false positives and prompts many skiers to disable the feature to avoid unnecessary alerts [8]. Therefore, to accurately detect falls in freestyle skiing, it is imperative to develop a more advanced and sophisticated system that takes into account the specific movements and forces associated with the sport. By addressing these unique challenges in the system's design, false positives can be minimised, resulting in a more precise and reliable fall detection system for freestyle skiers.

To address the challenges of accurate fall detection in freestyle skiing, this project implements and compares multiple deep learning models to classify between fall and non-fall events. Deep learning models were selected due to their widespread use and success within other high-speed fall detection domains, such as cycling, sport climbing and running [9, 10, 11]. Moreover, deep learning models were preferred over other classification algorithms due to their ability to automatically extract features and capture complex patterns within data. This enables them to effectively handle the presence of noise caused by the regular impacts and rotations inherent in skiing. The resulting system consequently achieves higher levels of accuracy by minimising the occurrence of false positives.

1.2.1 Objectives and Requirements

For this project to be considered successful, there are a number of objectives, requirements and success criteria that must be fulfilled, as outlined below:

- Determine an appropriate dataset and data format for analysis (Section 1.3 and 2.2)
- Preprocess the raw data for more effective and efficient training and analysis (Section 2.1)
- Develop a variety of deep learning models utilising different model architectures, training processes and input types (Sections 2.3-3.3)
- Compare each model's effectiveness using the evaluation metrics outlined below (Section 4)

Evaluation Metrics:

$$Recall = \frac{TP}{TP + FN}$$ proportion of true falls correctly classified as falls

$$Precision = \frac{TP}{TP + FP}$$ proportion of samples classified as falls that are true falls

$$F1\ Score = \frac{2 * (Precision * Recall)}{Precision + Recall}$$ harmonic mean of precision and recall to combined

These performance metrics were selected because a successful fall detection system should not only ensure a user's safety by correctly classifying true fall events (high recall), but also ensure events classified as falls are indeed true falls (high precision), thereby minimising the time wasted in unnecessarily alerting medical services. [7]. Therefore, to measure and compare the performance of each deep learning model, precision, recall, and their combined F1 score were used. Additionally, to aid in visualising the performance of each model, a confusion matrix was utilised to offer a quick overview of the aspects of the data that the model fails to predict correctly.

1.3 Datasets

As there is limited research on the use of fall detection systems in freestyle skiing, there are no datasets publicly available for training deep learning models to classify falls within the sport. Therefore, as part of this study, a dataset was collected that contains tri-axial accelerometer and gyroscope readings from skiers performing a variety of tricks as well as unintentional falls. Whilst some studies utilise accelerometers alone, these often result in lower precision and recall due to the similar acceleration patterns present in fall and near-fall events. This project therefore also incorporates gyroscope readings to measure changes in orientation, as they have been proven to aid in the accurate differentiation of falls and other impacts [7]. Additionally, to enhance the convenience and cost-effectiveness of data collection, the built-in accelerometers and gyroscopes present in modern smartphones were used [12, 13]. This is achieved through the 'Sensor Logger' application available on both iOS and Android phones that has demonstrated high levels of accuracy within other fall detection studies [14, 15].

The placement of sensors varies considerably across different research projects. However, sensors placed on the hips or waist often result in the highest performance metrics, likely because they experience less movement within normal activity than other body parts. This reduces noise in the sensor readings, resulting in the easier detection of falls [16, 17, 18]. As a result, participants in this study were instructed to carry their phones in the top pocket of their salopettes, positioned near their hip, to ensure consistent and uniform data readings across all participants.

Other Datasets Used:

Whilst a large number of samples can be efficiently collected, the process of labelling the dataset for supervised learning is a time-consuming task, requiring manual annotation of each individual sample as either a fall or non-fall event. Therefore, labelling the entire dataset is not feasible, and results in only a small subset of samples being labelled. Transfer learning, amongst other methods, was therefore utilised to determine if knowledge learnt from similar tasks can be applied to accurately predict the collected data. In these methods, the SDSU fall dataset was utilised [7, 19]. This dataset contains tri-axial accelerometer and gyroscope readings from sixteen subjects, with sensors placed on various body parts, capturing a diverse range of falls and non-falls. Unlike other similar datasets, SDSU also contains near falls and ADLs (activities of daily life) such as walking, running, jumping, and sitting down. This creates a more varied dataset likely to result in a model that generalises better to real world situations such as freestyle skiing. Additionally, the dataset includes labels that define the start and end of a fall. This facilitates easy pinpointing of falls within a sequence and efficient labelling of sub-sequences created by sliding window segmentation (discussed in Section 2.1).

2 Image-Based Transfer Learning

The first methods tested in this project focus on converting the raw accelerometer and gyroscope signals into 2D images for analysis by deep neural networks to classify fall and non-fall events. This section outlines the steps, including data pre-processing, image transformation, and deep learning models used to develop such a system.

2.1 Pre-processing

Before the accelerometer and gyroscope signals can be used to train the fall detection models, the data must be pre-processed. Data pre-processing is a crucial step in preparing datasets for deep learning networks, involving cleaning, transforming, and organising data into a format that can be more easily and effectively analysed [20]. This process enhances the quality and relevance of information input into the neural network, facilitating more effective learning and decision-making processes to ensure optimal performance and accurate predictions.

Low Pass Filtering

The first pre-processing step utilised is low pass filtering, with the aim of reducing noise within the dataset. This process removes short-term fluctuations in the sensor's readings whilst preserving the overall structure of the signal [7, 21]. Figure 2.1 below illustrates the outcome of this process, in which the original signal (depicted in blue) undergoes low pass filtering, reducing noise whilst maintaining the global pattern (depicted in orange).

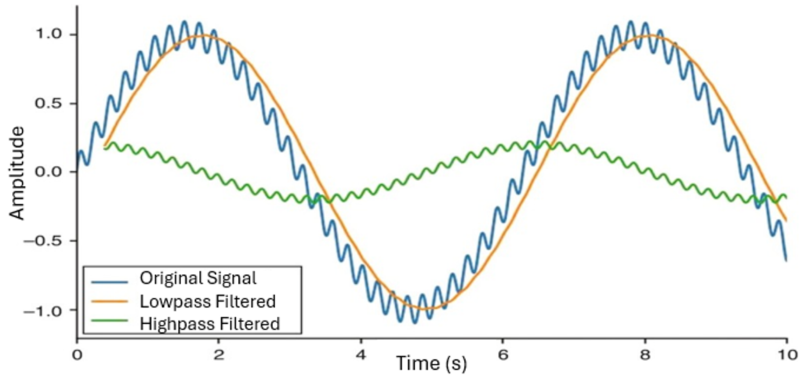


Figure 2.1: Illustration of Low Pass Filtering [22]

The resulting noise reduction aids in fall detection by enabling the system to prioritise significant changes in the data that are typically associated with a fall, whilst minimising the influence of minor fluctuations, such as movement of the phone sensors inside a pocket . This pre-processing technique has proven to be effective in enhancing the performance of other fall detection systems, resulting in improved accuracy and specificity in 80% of experiments compared to systems that use unfiltered data [21]. Furthermore, to detect falls in freestyle skiing, low-pass filtering helps to mitigate the effects of the variations introduced by the rapid movements and frequent impacts in the sport, increasing accuracy and specificity in classification.

Feature Normalisation

Feature normalisation is an essential pre-processing step used within deep neural networks that ensures all input features are within a similar range and prevents biases between features. As the accelerometer and gyroscope readings are likely to have different initial ranges, normalisation ensures both sensors contribute equally to the learning process, resulting in more accurate classification of falls [23].

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

The project utilised min-max normalisation (equation shown above) to transform the original data into a range of 0 to 1. Although other forms of normalisation are available, such as Z-score normalisation and mean normalisation, min-max was selected due to its widespread use and success in other fall detection systems [7, 24, 23].

Sliding Window Segmentation

The SDSU fall dataset and the dataset collected in this project both contain time series sequences of varying lengths, with falls and other ADL events occurring at different times in each sample. However, the deep learning models used to classify each sample require all inputs to be of a fixed, pre-specified length that varies depending on the model used. Therefore, each sequence is segmented into smaller fixed length segments that are able to be processed by the fall detection models.

Sliding window segmentation divides time series data by incrementally shifting a fixed-length window by a predetermined stride (the step size between windows), defining a new sample at each iteration. There are two main types of sliding window segmentation: non-overlapping, where each window has no intersection with others, and overlapping, where windows have a degree of overlap with adjacent windows [7]. Whilst non-overlapping is efficient, as it avoids repetition of information, an issue arises when important fall data is split between two windows, often leading to the system incorrectly labelling these falls [25]. Therefore, this project utilises overlapping windows to increase the likelihood of capturing entire fall sequences in a sample by repeating information at the edges of each window, whilst also increasing the number of data samples created to train the model.

A window size of 4 seconds, with an overlap of 1.5 seconds, was selected for the image-based learning methods discussed in this section. This was based on the observation that the majority of falls in the datasets last between 2.5 and 4 seconds. Therefore, by selecting a window size of 4 seconds, all informative fall data can be captured within a single window without the inclusion of irrelevant information that may reduce the accuracy of classification. Additionally, windows were labelled as falls if 5% or more of the sequence lies between the dataset's labelled fall segments.

Dataset Rebalancing

Both datasets used in this project consist of predominantly non-fall events, with only 13% and 20% of samples in the SDSU and collected dataset labelled as falls respectively. However, class imbalances in training data often result in models that place too much focus on the majority class. As a result, the model neglects the importance of the minority class and reduces the accuracy of predicting true fall samples, leading to a system that fails to correctly identify falls [18]. Therefore, this project utilised dataset resampling methods before training the models to mitigate the impact of these issues.

The most commonly used resampling methods are random oversampling and random undersampling. Random oversampling involves rebalancing a dataset by duplicating samples of the minority class within the training set. While this technique can address class imbalance, the duplicated samples do not provide new information to the model and can increase the risk of overfitting to the training data. Consequently, this reduces the system's ability to generalise to new data and reduces the accuracy of predictions for new, unseen samples. In contrast, random undersampling involves randomly removing samples from the majority class in the training set to rebalance the dataset. However, this approach results in the loss of potentially valuable information during training, which may lead to less accurate classification [7, 18].

A common approach to counteract the issues associated with oversampling and undersampling is to combine both methods. This approach reduces the number of duplications and removals required to balance the training sets, thus minimising the risks of overfitting and information loss. Such an approach has proven effective within other imbalanced classification models, increasing both recall and precision [26]. Consequently, this project utilised both random oversampling and random undersampling techniques to improve the accuracy of predicting minority class of true fall events.

2.2 Image Creation

To convert the time series windows into 2D images, this project utilised Gramian Angular Fields (GAF). Gramian Angular Fields are a mathematical tool used to convert time series data into 2D images, first by converting the series into polar coordinates and then calculating the pairwise dot products of all time steps. This results in a matrix of values between -1 and 1 that is visualised as a heatmap and captures both the temporal patterns and the underlying structure of the data. Each row and column in the matrix corresponds to a time step, with its values representing the correlation to all other time steps within a window. Values close to 1 denote a positive correlation of time steps, whilst values close to -1 represent a negative correlation [7, 27]. The Gramian Angular Field for a single feature in an example fall sample is displayed in image (a) of Figure 2.2 below.

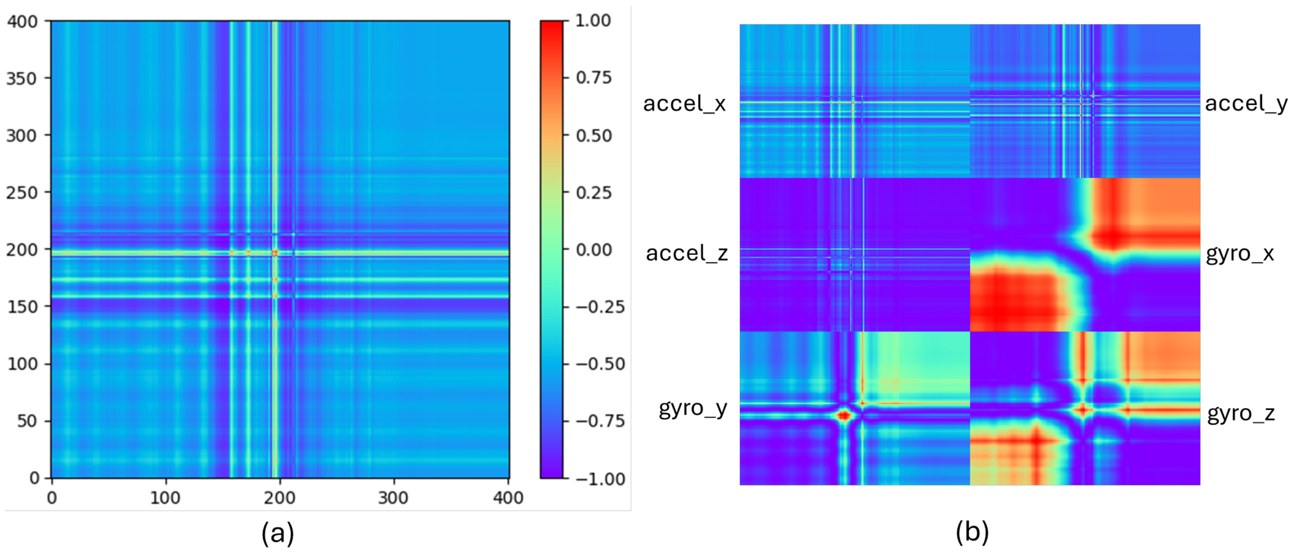


Figure 2.2: Example Gramian Angular Field and Multi-Variate Concatenation

GAF was chosen for this project due to its proven success in visualising time series data within other fall detection systems and time series classification studies, where it has demonstrated high recall and precision compared to alternative visualisation methods [27, 28]. Furthermore, GAF offers a significant advantage over other time series visualisation techniques by preserving the temporal dependencies crucial for identifying a fall, ensuring accurate prediction of fall and non-fall events [7, 29]. However, each heatmap generated from the GAF matrices represents a single feature of the data in each window sample. Given that each window contains six features captured by tri-axial readings of both the accelerometer and gyroscope, the information from all six heatmaps must be combined into a single image for analysis by the deep neural networks. This is achieved by concatenating the six heatmaps into a single image, as utilised in other time series classification studies [30, 31]. An example image containing all six features is displayed in image (b) of Figure 2.2.

2.3 Convolutional Neural Networks

With the accelerometer and gyroscope datasets pre-processed and visualised, convolutional neural networks (CNNs) were utilised to analyse and classify the GAF of each sample to distinguish between fall and non-fall events. CNNs are a widely used type of deep learning network specifically designed to process images and are renowned for their ability to automatically extract the features relevant for accurate prediction [32]. This is achieved through the use of convolution layers that utilise filters to extract local features within images such as repeated shapes, colours and edges. By combining filters within multiple layers in the network, more complex features are derived to create a final prediction for each sample [7].

To learn the features important for classification, CNNs utilise backpropagation to adjust the network's weights and minimise the error between the predicted and true labels during training. However, training a CNN from scratch is time-consuming and requires an exceptionally large dataset not typically available for tasks such as fall detection. Therefore, it is common practice to pre-train a CNN on a sufficiently large, unrelated dataset, save the weights learnt, and then fine-tune the network on the relevant dataset. This technique, known as transfer learning, allows the network's early layers, responsible for extracting low-level features such as corners and edges, to be pre-trained on a large, unrelated dataset. The last few layers of the network are then retrained on the limited dataset to extract the high-level features, such as textures and objects, specific to the task at hand [33, 34, 35]. Transfer learning, therefore allows for accurate classification models to be trained when the dataset available are insufficient in size.

Two high-performing and commonly used CNN architectures, AlexNet and ResNet50, were implemented and compared for the image based analysis in this project. Both architectures were pre-trained on the ImageNet dataset that contains millions of human annotated images from a variety of categories such as mammals, vehicles and flowers. Both architectures were then fine-tuned on the SDSU and collected fall datasets and their performance on the collected dataset assessed using the evaluation metrics outlined in Section 1.2.1.

2.3.1 AlexNet

AlexNet is a frequently utilised deep CNN that first emerged in 2012 when it demonstrated unparalleled performance in analysing the ImageNet dataset during the LSVRC-2012 competition, outperforming the second place network's accuracy by 10.9% [36]. As illustrated in Figure 2.3, AlexNet consists of five convolutional layers, responsible for feature extraction, and three fully connected layers to map the extracted features to the output classes.

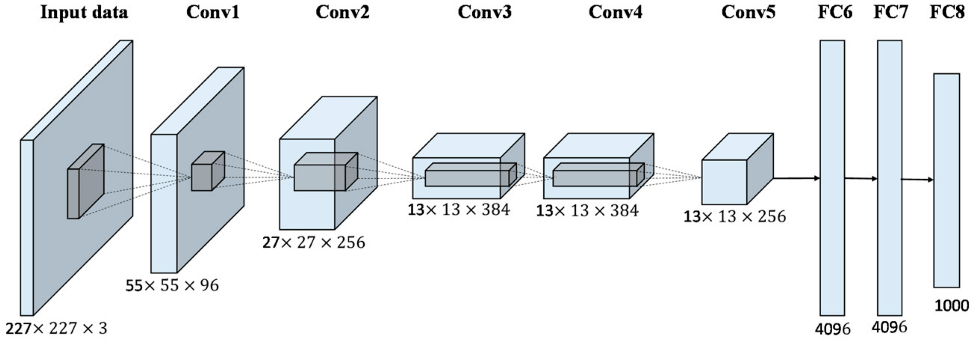


Figure 2.3: Illustration of AlexNet's Architecture [37]

AlexNet is designed to process 3-channel RGB images with dimensions of 227 pixels in both height and width, resulting in images of size 227x 227x3. The GAF images are therefore resized to fit this requirement before they are analysed by the network [7]. However, before utilising AlexNet to classify falls, some minor adjustments were made to its architecture to enhance its suitability for use in this project. Specifically, the number of nodes in the last fully connected layer was modified from 1000 to 1, with a sigmoid activation function to calculate the final binary class prediction as either a fall or a non-fall. Whilst AlexNet typically uses a softmax function for the final prediction, sigmoid was chosen due to its suitability for binary classification tasks, offering increased computational efficiency and performance when distinguishing between two classes [38]. Similarly, binary cross-entropy was selected as the loss function during training as it is specifically designed for binary classification problems, often resulting in faster convergence and the development of robust models that generalise effectively to new data. Additionally, during testing, it was found that assigning class weights inversely proportional to the number of samples within each class helped overcome the imbalanced dataset. This enabled the model to focus more on the minority class of falls, thus reducing the number of false positives and false negatives. Consequently, weighted binary cross entropy was utilised as the loss function utilised during training.

The implementation of transfer learning using a pre-trained AlexNet network is widely used in image analysis. However, studies vary in the selection of which layers are further trained and which are frozen to preserve pre-trained weights. A common approach is to freeze the convolutional layers and only fine-tune the weights of the three fully connected layers at the end of the network, as the features extracted by the convolutional layers are often general and transfer well between different tasks and datasets. Therefore, fine-tuning of the fully connected layers alone can adapt the network to the required task and reduce the time taken to accurately train the model [7, 39]. Some studies suggest that further training the entire network may lead to more accurate predictions as fine-tuning the convolutional layers enables them to adapt to the target domain and identify the features more relevant to the end classification. However, this approach increases the risk of overfitting to the training data, especially when the target dataset is limited, and may reduce the accuracy of predicting the new, unseen test samples [35]. Therefore, the following transfer learning configurations were implemented to determine the optimal layers for fine-tuning the network. The results of each configuration are presented and evaluated in Section 4.

(A): Retrain the whole network, altering the transferred weights for all layers in the network.

(B): Retrain only the fully connected layers, freezing the previous convolutional layers.

(C): Retrain only the last layer, freezing all previous layers.

2.3.2 ResNet50

The second pre-trained network implemented to classify the Gramian angular field visualisations of falls was ResNet50. ResNet50 is a commonly used CNN designed to address the vanishing gradient problem that occurs in deep learning models. This issue arises due to the repeated multiplication of weights during backpropagation, that results in increasingly small gradients calculated. Consequently, the early layers of the network receive progressively smaller gradients, significantly slowing down the training process and impairing the network's ability to extract relevant features for classification thereby reducing the model's accuracy [40, 41].

To address the vanishing gradient problem, ResNet introduced residual blocks with skip connections. These blocks contain multiple convolutional layers and utilise skip connections to feed the output of a layer as the input of a later layer, skipping one or more intermediate layers [7]. Skip connections therefore provide an alternative path for the gradient to flow, reducing the number of gradient multiplications required and preventing the gradient from vanishing during backpropagation. This enables the training of deeper networks capable of capturing more complex patterns, leading to higher accuracy across a wide range of image analysis tasks [40, 42]. Consequently, ResNet is likely to better differentiate between the patterns that distinguish intentional manoeuvres from unexpected falls, creating a system capable of accurately detecting falls within freestyle skiing.

The typical architecture of ResNet50 includes an input layer of size 224x224x3, a convolutional layer, 16 residual blocks, and a fully connected layer [43]. However, before utilising ResNet50 for fall classification, some minor adjustments are necessary to adapt the network to classify GAF images. As with AlexNet, these modifications involve altering the fully connected layer to a single node with a sigmoid activation function, and utilising weighted binary cross entropy as the loss function during training to represent the binary classification for detecting falls. Additionally, similar to the transfer learning approach applied to AlexNet, identification of the optimal layers for fine-tuning is essential for maximising the performance of ResNet50. Therefore, the following configurations were implemented, with their results presented and evaluated in Section 4.

- (A): Retrain the whole network, altering the transferred weights for all layers in the network.
- (B): Freeze the convolutional layer and early residual blocks that capture the low-level features likely to transfer well to new datasets. Retrain the last 3 residual blocks and the fully connected layer to capture the high-level features specific to the fall dataset.
- (C): Retrain only the fully connected layer

3 Raw Signal-Based Learning

The final fall detection methods tested in this project utilise deep neural networks to analyse raw accelerometer and gyroscope signals. Two commonly used deep learning models were implemented in this section, with both models being pre-trained on the SDSU fall dataset to help account for the limited labelled freestyle skiing samples collected. Both models were subsequently fine-tuned and tested on the collected dataset to evaluate their effectiveness in detecting falls within freestyle skiing.

3.1 Pre-processing

The data pre-processing methods used to analyse the raw signals are consistent with those previously discussed for image-based analysis. These methods consist of low-pass filtering, min-max normalisation, sliding window segmentation, and dataset rebalancing. However, a notable difference lies in the size of the window selected to segment the samples.

The GAF transformation used within the image-based methods compresses the temporal information of each feature to a fixed-size 2D representation processed by deep networks. This allows for the use of larger window sizes capable of capturing more data relevant for fall classification, without significantly increasing computational complexity. However, the six features (x, y, z for accelerometer and gyroscope) utilised in the raw signal-based methods, each sampled at a rate of 100Hz, create a considerable volume of data per second. As a result, the use of larger window sizes within these methods leads to high computational complexity during training. This slows down the convergence rate and also increases the risk of issues, such as the vanishing gradient problem, that reduce the accuracy of the system [41, 44].

For these reasons, it is important to select a window size that contains sufficient information to classify falls whilst minimising the computational complexity. In this project, an initial window size of 3 seconds with an overlap of 1 second was chosen for the raw signal-based learning methods. This decision was based on its utilisation in other fall detection studies that have demonstrated high levels of accuracy, specificity, and recall [7, 45]. However, adjustments were made to this value throughout the implementation and experimentation of the networks to help optimise the performance of the fall detection system.

3.2 Recurrent Neural Networks and Long Short-Term Memory

Recurrent Neural Networks (RNNs) are a specialised type of deep learning network, specifically designed to process sequential data. Unlike traditional deep learning models, RNNs incorporate loops within the model's layers to utilise previous information in the analysis of future data within a sequence [7, 41]. This grants them the ability to analyse sequences as a whole and learn the complex patterns that emerge within sequential data. However, the repeated multiplications in these loops during backpropagation also increase the risk of the vanishing gradient problem and decreases the network's ability to learn long-term patterns, reducing the accuracy of the classification.

Long Short-Term Memory (LSTM) models are a subset of RNNs that address the vanishing gradient problem by introducing a memory component known as the cell state. This cell state contains three 'gates' that control the flow of information, allowing the model to selectively remember or forget information over long sequences. By retaining only the information relevant for classification, LSTMs can capture long-term dependencies in the data whilst reducing the risk of the vanishing gradient problem. Consequently, LSTMs are widely employed in fall detection systems, often achieving high levels of accuracy, specificity, and recall. Therefore, within this project, LSTMs were utilised to classify the raw sensor signals, with the complete network architecture outlined below:

1. Two LSTM layers, each with 128 nodes
2. A dropout layer to prevent overfitting
3. Two fully connected layers containing:
    (a) 128 nodes and ReLU activation
    (b) 1 node and sigmoid activation to create the final prediction

The number of layers and nodes utilised in LSTMs varies between studies, although the overall structure remains fairly consistent, containing one or more LSTM layers, dropout regularisation and fully connected layers. This is reflected in the proposed architecture above, with two LSTM layers containing 128 nodes selected based on findings within other fall detection studies. These studies have shown that networks containing two LSTM layers with 128 nodes achieve the highest accuracy within a relatively short training time [7, 46].

Additionally, dropout regularisation involves setting a small portion of nodes to zero during each training batch. This introduces randomness during training, reducing the risk of overfitting and enhancing the model's ability to generalise to new samples. However, setting too many nodes to zero may hinder the network's ability to learn meaningful patterns within the data, ultimately reducing the classification accuracy [46]. Therefore, various dropout rates ranging between 0.01 and 0.5 were tested to find the optimal rate that reduces overfitting without decreasing accuracy.

3.3 CNN-LSTMs

The final network architecture implemented to analyse the raw sensor signals was a CNN-LSTM. This is a hybrid architecture that implements elements of the previously discussed CNN and LSTM architectures to create a network that combines the benefits of both architectures [47]. The resulting network utilises 1 dimensional convolutional layers to extract the relevant features present within the sequential sensor signals. These extracted features are subsequently used as input to the LSTM, reducing the length of the sequences analysed by the LSTM. This reduces the risk of the vanishing gradient problem and often results in better generalisation to new samples, increasing the accuracy of the test set [48, 49].

Combining CNN and LSTM architectures is a commonly utilised approach that has proven effective in other fall detection studies [7, 48, 49]. Therefore, this study implemented a CNN-LSTM hybrid architecture to analyse and classify the time-series sequences captured within freestyle skiing falls. The network architecture utilised to achieve this is a modified version of the network described in Section 3.2, similar to those utilised within previous high performing fall detection systems [49, 50]. The full network architecture utilised is outlined below:

1. Two 1 dimensional convolutional layers
2. Max pooling and dropout
3. One LSTM layer with 128 nodes
4. Two fully connected layers containing
   (a) 128 nodes and ReLU activation
   (b) 1 node and sigmoid activation to create the final prediction

.

4 Results, Evaluation and Testing

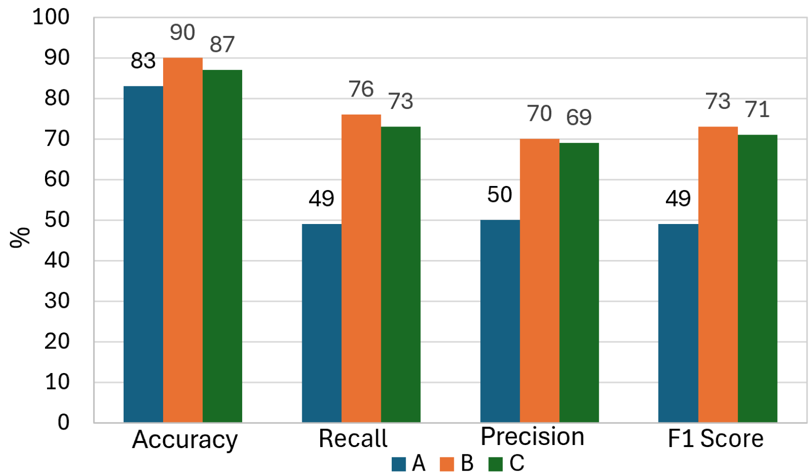
Using the methods previously described, the accelerometer and gyroscope signals were pre-processed and analysed using the four deep neural network architectures discussed in Sections 2 and 3. This section presents and evaluates the results for each architecture, utilising the evaluation metrics outlined in Section 1.2.1 to assess the ability of each network to correctly classify falls within freestyle skiing. The results are split into two main sections presenting results of the image-based and raw signal-based methods, together with a discussion of the advantages and limitations of each method.

4.1 Image-Based Classification

Using the methods outlined in Section 2 the pre-trained AlexNet and ResNet50 networks were initially further trained using the publicly available SDSU fall dataset. The networks were subsequently fine-tuned and tested on the collected freestyle skiing fall dataset, divided into 80% training and 20% testing sets. Training epochs ranged from 0 to 20, with the displayed results reflecting the highest performing epoch. The results for both pre-trained architectures are provided in their respective subsections, and provide analysis into the optimal transfer learning strategy to maximise their effectiveness in detecting falls within freestyle skiing.

AlexNet

Figure 4.1 and Table 4.1 present the average results of the previously discussed transfer learning configurations for AlexNet (Section 2.3.1), with the best results highlighted in grey within the table. These configurations represent the selection of pre-trained layers that are further trained or frozen and are labelled A, B and C as follows: (A) further trains all layers, (B) further trains only the fully connected layers, and (C) further trains only the final fully connected layer.



	Accuracy (%)	Recall (%)	Precision (%)	F1 Score (%)
A	83	49	50	49
B	**90**	**76**	**70**	**73**
C	87	73	69	71

Figure 4.1, Table 4.1: AlexNet Transfer Learning Results

As illustrated in Figure 4.1 and Table 4.1, configuration A (displayed in blue) yields the lowest scores across all four performance metrics. Despite achieving an accuracy of 83%, the relatively low recall, precision and F1 values of 49%, 50% and 49% respectively, highlight limitations in correctly identifying true fall samples. These limitations are likely caused by a loss of generalisation introduced by adjusting the weights of all network layers during training. Consequently, the model overfits to the training data, reducing its ability to correctly classify new, unseen samples.

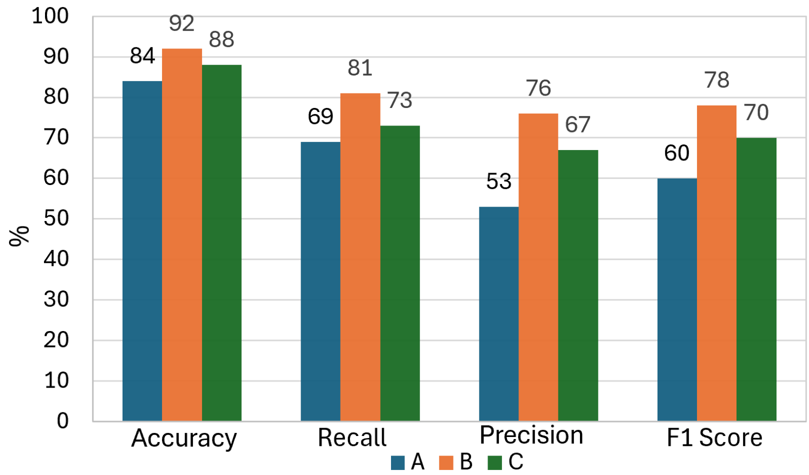
Configuration B (displayed in orange) is the best performing AlexNet model, achieving the highest scores across all performance metrics. By further training only the fully connected layers whilst retaining the convolutional weights from the pre-trained network, the network can adapt the pre-trained extracted features to the classification of falls without overfitting to the training set. This enables the model to learn the underlying patterns that differentiate fall and non-fall samples, without capturing the noise present in the training data. Consequently, this approach minimises the number of false positives and false negatives, as demonstrated by the improved recall and precision values of 76% and 70% respectively.

Configuration C (displayed in green) also demonstrated competitive performance, with scores across all performance metrics between only 1% and 3% lower than configuration B. These results indicate that further training of only the last fully connected layer effectively utilises the pre-trained weights for accurate fall classification. However, the small decrease in performance compared to configuration B can likely be attributed to the limited adaptability provided by altering only the last layer.

Consequently, the model may not fully capture the intricate patterns that differentiate between fall and non-fall samples, reducing the accuracy of predictions. However, the fine-tuning of only a single layer, containing 4097 trainable parameters, reduces the computational resources required to train the model compared to the trainable 54,538,241 parameters required by configuration B. Therefore, despite the small difference in performance, configuration C presents a viable alternative, particularly in scenarios where computational resources or training time are limited.

ResNet50

Figure 4.2 and Table 4.2 present the average results for the three previously discussed transfer learning configurations applied to ResNet50, with the best results highlighted in grey in the table. Similar to AlexNet, these configurations are labelled A, B and C as follows: (A) further trains all layers, (B) further trains the last three residual blocks and fully connected layer, and (C) further trains only the final fully connected layer.



	Accuracy (%)	Recall (%)	Precision (%)	F1 Score (%)
A	84	69	53	60
B	**92**	**81**	**76**	**78**
C	88	73	67	70

Figure 4.2, Table 4.2: ResNet50 Transfer Learning Results

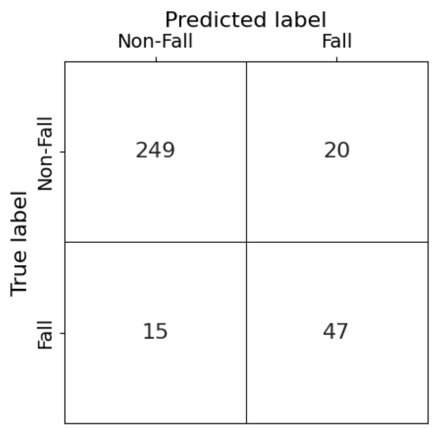
The results of ResNet50 presented in Figure 4.2 and Table 4.2 exhibit a similar pattern to the previously discussed results for AlexNet. Configuration A (displayed in blue) yields the lowest results across all four performance metrics, likely due to overfitting introduced by further training all layers within the network. This decreases the accuracy in predicting the test set, resulting in a higher number of false positives and false negatives, as demonstrated by the relatively low recall and precision of 69% and 53% respectively.

Configuration B (displayed in orange) emerges as the best performing model, achieving an accuracy of 92% with a recall and precision of 81% and 76% respectively. Similar to the results obtained for AlexNet, retaining the pre-trained weights of the earlier layers, responsible for extracting low-level general features, whilst further training the last 3 residual blocks and fully connected layer, enables the pre-trained network to adapt to classifying falls without overfitting to the training data. Therefore, the accuracy of predicting the unseen test set is increased.

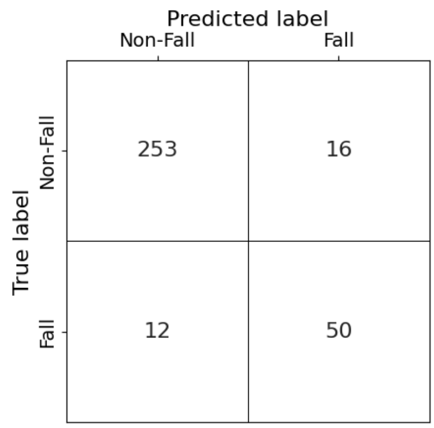
Configuration C (displayed in green) also demonstrates relatively high performance, obtaining an accuracy of 88% and values of 73%, 67% and 70% for recall, precision and F1 scores respectively. However, the performance gap between B and C is more significant than the difference observed within AlexNet. Notably, configuration C experiences a notable decrease of 9% in precision, whereas AlexNet only experiences a 1% decrease. This decrease in performance is likely due to ResNet50 being a much deeper network, containing 50 layers compared to AlexNet's 8 layers. Consequently, further training the last layer alone may not capture the relevant features for fall detection, reducing the accuracy of predictions.

Comparative Analysis and Benchmarks

The image-based transfer learning methods used in this project have proven effective in accurately classifying falls within freestyle skiing. This is demonstrated by the high levels of accuracy, precision, recall, and F1 scores achieved by both AlexNet and ResNet50. Furthermore, the use of transfer learning addresses the issue of overfitting associated with limited datasets, such as the collected freestyle skiing fall dataset. This enables the development of fall detection models that effectively generalise to new data, thereby minimising the number of false positives and false negatives in predicting the unseen test set. High values of precision and recall are therefore attained by both networks. This is further illustrated in Figure 4.3 below, that contains the confusion matrices for one run of configuration B for both AlexNet and ResNet50 models, providing a visual comparison between the predicted and actual labels of the test set.



(a) AlexNet



(b) ResNet50

Figure 4.3: Comparison of Confusion Matrices

These confusion matrices visualise the effectiveness of the image-based transfer learning methods in correctly distinguishing between fall and non-fall samples. Confusion matrix A shows that AlexNet correctly identified 47 out of 62 falls and 249 out of 269 non-falls. Confusion matrix B illustrates slightly better performance achieved by ResNet50, correctly identifying 50 out of 62 falls and 253 out of 269 non-falls, resulting in higher performance metric values as previously presented in Figure 4.2 and Table 4.2.

As deep neural networks operate as black box algorithms, the inner workings of the network and the extracted features are difficult to interpret. Consequently, the reasons for ResNet50's increased performance over AlexNet remain unclear. However, this improvement can likely be attributed to two main advantages offered by ResNet50's architecture. Firstly, ResNet50 is a deeper network, containing

50 layers compared to AlexNet's 8. This depth enables the network to learn more complicated patterns in the data, making it better suited for tasks with high intricacy such as detecting subtle differences in GAF images between falls and non-falls. Secondly, the skip connections present within ResNet50 help to mitigate the vanishing gradient problem, enabling the efficient training of deeper networks and increasing the accuracy of predictions.

No experiments have yet been performed on the use of deep learning fall detection methods in freestyle skiing. Consequently there are no studies available to which the results from this project can be directly compared and contrasted. Results can, however, be compared to those available in fall detection systems developed for elderly care. In this respect, a paper by Qi et al investigates the classification of falls in the elderly by the analysis of GAF images and reports an accuracy of 95% [28], comparing closely to 92% accuracy achieved by ResNet50 in this project. However, the study by Qi et al produced values of 94% and 97% for precision and recall respectively and are in contrast to the lower levels of 76% and 81% obtained in this project. This suggests that whilst the image-based methods used in this project correctly identifies most samples, there is a relatively higher proportion of false positives and false negatives when compared to those observed in fall detection in the elderly. These differences are to be expected as the high speeds, rotations and frequent impacts within freestyle skiing introduces significant variability that is not present within elderly fall detection datasets. Therefore, falls become more challenging to distinguish from usual activity in the sport, leading to an increased number of false positives and false negatives. Fall detection systems developed for use in elderly care therefore offer a limited, yet helpful, comparison that can be used to determine limitations of the methods implemented. This will assist in the development of improved methods in future studies, as discussed in Section 5.

4.2 Raw Signal-Based Classification

Using the methods outlined in Section 3, the LSTM and CNN-LSTM were trained using the publicly available SDSU fall dataset. Subsequently, they were fine-tuned and tested on the collected freestyle skiing fall dataset, divided into 80% training and 20% testing sets. Training epochs ranged from 0 to 20, with the displayed results in Table 4.3 reflecting the highest performing epoch for each network, as identified by the performance metrics discussed in Section 1.2.1.

	Accuracy (%)	Recall (%)	Precision (%)	F1 Score (%)
LSTM	87	61	66	63
CNN-LSTM	**91**	**78**	**73**	**75**

Table 4.3: LSTM and CNN-LSTM Results

The results presented in Table 4.3 demonstrate that the utilisation of a traditional LSTM, as discussed in Section 3.2, struggles to accurately distinguish between fall and non-fall samples. Despite achieving an accuracy of 87%, the values obtained for recall, precision, and F1 of 61%, 66%, and 63% respectively are relatively low, indicating a significant number of false positives and false negatives. Moreover, the training process was often observed to be unstable, with the binary cross entropy fluctuating between each epoch, suggesting that the model struggled to identify meaningful patterns for the correct classification of falls.

As LSTMs are black box algorithms, pinpointing the exact reasons for training instability, low recall, and low precision is challenging. However, unlike the image-based methods, the LSTM model implemented is not pre-trained, making it prone to overfitting to the limited dataset collected. Consequently, the model learns the noise within the training set rather than the patterns that distinguish falls and non-falls within freestyle skiing, thereby decreasing the accuracy of predictions. Efforts were made to mitigate overfitting by adjusting the network previously described in Section 3.2. This included experimenting with various configurations, such as different numbers of LSTM layers, varying the number of nodes within each layer, and dropout rates ranging from 0.1 to 0.5. However, these adjustments had minimal impact on the results, with the best results (displayed in Table 4.3) obtained for two LSTM layers each with 128 nodes and a dropout rate of 0.5.

The CNN-LSTM model, previously discussed in Section 3.3, demonstrated superior performance compared to the LSTM, achieving an accuracy of 91% along with increased values of recall, precision, and F1, reaching values of 78%, 73%, and 75%, respectively. This suggests that the addition of convolutional layers to the network helps overcome the issues associated with traditional LSTMs, creating a model that can be trained effectively to accurately classify falls. As with the results for the previously discussed methods, the black box nature of neural networks makes it difficult to identify the exact reasons for this increased performance. However, it can likely be attributed to several key benefits provided by the addition of convolutional layers.

Firstly, the feature extraction of convolutional layers reduces dimensionality in the data, enabling the model to filter out noise and prioritise the information important to the classification [51]. Consequently, this reduces the risk of overfitting and increases the accuracy of predictions. Secondly, this dimensionality reduction also reduces the risk of the vanishing gradient problem, ensuring the effective analysis of the long sequences present within fall detection. Finally, convolutional layers are capable of learning hierarchical features, with the initial layers learning low-level features that are subsequently combined by deeper layers to enable the extraction of more complex patterns than an LSTM alone. These benefits therefore create a high performing model that is able to accurately classify falls with minimal false positives and false negatives, as displayed by the high values of recall and precision in Table 4.3.

Comparative Analysis and Benchmarks

Overall, the raw signal-based methods utilised in this project obtained varying results. Traditional LSTMs encountered issues during the training process, leading to a model that struggles to differentiate between the true fall samples and non-falls, as demonstrated by the relatively low values of recall and precision presented in Table 4.3. However, by combining CNNs and LSTMs through the addition of feature-extracting convolutional layers into an LSTM architecture, these issues were minimised. The resulting hybrid model was therefore able to identify and extract the features relevant for classification, thereby reducing the number of false positives and false negatives and consequently improving the recall and precision values obtained.

As previously discussed within the image-based methods, the lack of research assessing the use of fall detection methods within freestyle skiing means there are no studies to which this project's results can be directly compared and contrasted. Consequently, comparisons are instead drawn from studies within other domains such as fall detection in the elderly. A study by Tsinganos and Skodras (2017) that similarly utilised data collected from smartphones reports an accuracy of 97.5% when detecting falls within the elderly, close to the 91% obtained by the CNN-LSTM in this project. However, the paper by Tsinganos and Skodras (2017) also reports higher values of 95.5% and 92% for recall and precision respectively, compared to the 78% and 73% obtained by the CNN-LSTM in this project [13]. As discussed within the image-based methods, it is important to consider the differences between the datasets utilised when making these comparisons. The increased movement and regular impacts inherent within freestyle skiing are likely to introduce errors into the model's predictions. Consequently, fall detection systems within other domains should not be used as a direct comparison but instead serve as references to identify areas for future work to improve the methods implemented in this project.

5 Discussion and Conclusion

Whilst both the image-based and signal-based methods utilised in this project were effective in accurately classifying falls and non-falls within freestyle skiing, their respective results reveal differences that highlight the advantages and limitations associated with each approach. This section addresses these advantages and limitations and their impact on the final fall detection model, along with potential future work and improvements to the methods implemented.

5.1 Performance Comparison

Both the image-based and signal-based methods achieved high levels of accuracy, with the highest performing image-based method (ResNet50) achieving an accuracy of 92%. Similarly, the highest performing signal-based method (CNN-LSTM) achieved an accuracy of 91%. However, there is a larger disparity within the recall, precision, and F1 values obtained for these methods. ResNet50 obtains values of 81%, 76%, and 78% for recall, precision, and F1, respectively, whereas the CNN-LSTM obtains 78%, 73%, and 75%. Whilst there is only a three percent difference across each metric, it does indicate that image analysis utilising ResNet50 reduces the number of false positives and false negatives compared to the sensor signal analysis utilising a CNN-LSTM. Due to the black box nature of these methods it is difficult to determine the exact reasons for this. However, the architectures and training procedures used by each method do provide some insight, such that the differences can likely be attributed to the following:

- **Transfer learning:** The ResNet50 network utilised in this project is pre-trained on a diverse dataset of images, which helps mitigate the risk of overfitting to the test set during further training. In contrast, the CNN-LSTM is not pre-trained, making it more susceptible to overfitting, thus increasing the likelihood of false positives and false negatives when predicting the unseen test set.
- **Reduced data complexity:** The transformation of time series data to GAF images reduces the complexity and dimensionality of the data provided as input to ResNet50. This both decreases the risk of overfitting and the vanishing gradient problem, thereby increasing the predictive capabilities of the network.
- **Deeper network:** ResNet50 is a significantly deeper network, incorporating skip connections that are not present in the CNN-LSTM architecture implemented. These skip connections mitigate the vanishing gradient problem, facilitating effective training of deeper architectures. Consequently, ResNet50 can learn more complex features that help to differentiate between falls and other impacts within freestyle skiing.

However, it's important to note that whilst ResNet50 achieves slightly higher performance metrics, it also comes with its own set of disadvantages. The GAF transformation of the time-series data required for the use of ResNet50 is computationally expensive and adds substantial overhead to the training and execution of the network. This is further compounded by the computational cost associated with ResNet50's deeper architecture compared to the CNN-LSTM implemented. Consequently, the training and inference times required by ResNet50 are significantly longer, taking around 28 times longer to train than the CNN-LSTM during the implementation of this project. Whilst this disparity in training time may vary depending on the hardware used, the overarching trade off remains consistent, with ResNet50's increased performance counterbalanced by its computational requirements.

Given the substantial computational resources and training time required, the viability of the ResNet50 depends on the hardware available and the requirements of the end system being developed. As a result, if computational resources are limited, the comparative performance of the CNN-LSTM may provide a more viable solution with minimal loss in the predictive capabilities of the fall detection system. Similarly, if a short response time is required and the hardware available is limited, ResNet50 may not offer a viable solution. In contrast, the efficiency of the relatively shallow CNN-LSTM architecture is likely to provide a much faster response. Consequently, the selection between ResNet50 and CNN-LSTM depends on a careful evaluation of the fall detection system's specific requirements, performance targets, available resources, and operational constraints.

5.2 Limitations and Future Work

Whilst the methods implemented showed considerable potential in detecting falls within freestyle skiing, a common issue of overfitting was often observed, increasing the number of false positives and false negatives in predicting the unseen test set. In a real world, deployed system these errors may have serious consequences. A high number of false positives, indicated by low precision, may result in wasted time by medical services as they respond to skiers who do not require assistance. Conversely, a high number of false negatives, indicated by low recall, may result in an insufficiently safe system, as help may not be provided to skiers who have experienced a fall. Consequently, if the current models were implemented into a real world application, it would be crucial to address this issue to minimise the number of false positives and false negatives obtained.

Due to the time taken to manually annotate each sample, the freestyle skiing dataset utilised within this project is relatively limited. This makes it more difficult to extract the relevant features and increases the influence of noise within each sample. Furthermore, the samples collected may not capture the full range of movements and intentional impacts that occur within freestyle skiing. As a result, patterns learnt from the training set may not apply to new data. Consequently, by training the deep learning networks on this limited dataset, they become more prone to overfitting.

Therefore, the collection of a larger and more varied dataset containing a diverse range of tricks from skiers of all abilities would likely be a priority within future work to improve the results obtained within this project. Additionally, ongoing monitoring and evaluation of the deployed system may facilitate continuous improvements, thereby enhancing the accuracy of the models over time. For example, implementing a user feedback system would enable skiers to report any undetected falls or false positives. This iterative improvement approach to the system's deployment would not only expand the size and diversity of the dataset but also help in identifying the correct changes to enhance the system's reliability and effectiveness in real-world settings.

Finally, during testing, the response time for the system to predict a single sample was notably fast, providing efficient and timely detection of falls. However, it's crucial to acknowledge that this performance may vary when the system is deployed on different platforms, such as mobile devices.

Factors such as hardware limitations and processing power can significantly impact response times, potentially leading to delays in real-time detection and analysis. Therefore, before deploying the methods implemented in this project, optimising the system for mobile platforms and conducting thorough performance evaluations across various devices is imperative to ensure consistent and reliable performance across different devices.

5.3 Conclusion

In conclusion, this project aimed to apply deep learning methods to analyse accelerometer and gyroscope sensor readings for classifying falls and non-falls within the high-speed sport of freestyle skiing. The methods implemented were divided into two main categories: image-based methods, which convert the sensor readings into Gramian Angular Field images before analysis, and raw signal-based methods that directly analyse the captured time-series sequences. Overall, both methods achieved high levels of accuracy, precision, recall, and F1 scores, indicating their effectiveness in automatically extracting features relevant to differentiate between falls and intentional movements within the sport. Subsequently, the advantages and limitations of each method are discussed, providing insight into the optimal choice based on the requirements and available resources of the end system, before presenting potential improvements and future work to address their limitations.

References

[1] International Olympic Committee. *History of Freestyle Skiing*. (last accessed: 19/11/2023). URL: https://olympics.com/en/sports/freestyle-skiing/.

[2] Nicolas Kurpiers. "Dynamics of Freestyle Skiing–Equipment development and implications for injury prevention strategies". PhD thesis. ResearchSpace@ Auckland, 2010.

[3] Nick Armes. "The Rise of Freestyle Terrain: Getting Extreme as an Increased Liability for Ski Resorts". In: *DePaul J. Sports L. & Contemp. Probs.* 9 (2012), p. 59.

[4] Tonje Wåle Flørenes et al. "Injuries among World Cup freestyle skiers". In: *British Journal of Sports Medicine* 44.11 (2010), pp. 803–808.

[5] Xue-Lei Fu et al. "Incidence of injuries in professional snow sports: a systematic review and meta-analysis". In: *Journal of sport and health science* 11.1 (2022), pp. 6–13.

[6] Xueyi Wang, Joshua Ellul, and George Azzopardi. "Elderly fall detection systems: A literature survey". In: *Frontiers in Robotics and AI* 7 (2020), p. 71.

[7] Haben Girmay Yhdego. "Wearable SensorGait Analysis for Fall Detection Using Deep Learning Methods". PhD thesis. Old Dominion University, 2023.

[8] Matt Richtel. "'My Watch Thinks I'm Dead'." In: *International New York Times* (2023), NA–NA.

[9] I-Hsuan Peng et al. "Development of a cycling safety services system and its deep learning bicycle crash model". In: *Journal of Communications and Networks* 24.2 (2022), pp. 246–263.

[10] Angelo Bonfitto et al. "Pattern recognition neural classifier for fall detection in rock climbing". In: *Proceedings of the Institution of Mechanical Engineers, Part P: Journal of Sports Engineering and Technology* 233.4 (2019), pp. 478–488.

[11] Taylor R Mauldin et al. "SmartFall: A smartwatch-based fall detection system using deep learning". In: *Sensors* 18.10 (2018), p. 3363.

[12] Stefano Abbate et al. "A smartphone-based fall detection system". In: *Pervasive and Mobile Computing* 8.6 (2012), pp. 883–899.

[13] Panagiotis Tsinganos and Athanassios Skodras. "A smartphone-based fall detection system for the elderly". In: *Proceedings of the 10th International Symposium on Image and Signal Processing and Analysis*. IEEE. 2017, pp. 53–58.

[14] Kelvin Choi. *awesome-sensor-logger*. (last accessed: 29/11/2023). URL: https://github.com/tszheichoi/awesome-sensor-logger.

[15] Sara Usmani et al. "Latest research trends in fall detection and prevention using machine learning: A systematic review". In: *Sensors* 21.15 (2021), p. 5134.

[16] Marvi Waheed, Hammad Afzal, and Khawir Mehmood. "NT-FDS—A noise tolerant fall detection system using deep learning on wearable devices". In: *Sensors* 21.6 (2021), p. 2006.

[17] Jiangpeng Dai et al. "PerFallD: A pervasive fall detection system using mobile phones". In: *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*. 2010, pp. 292–297. DOI: 10.1109/PERCOMW.2010.5470652.

[18] Jing Zhang, Jia Li, and Weibing Wang. "A class-imbalanced deep learning fall detection algorithm using wearable sensors". In: *Sensors* 21.19 (2021), p. 6511.

[19] Neuromechanics San Diego State University and neuroplasticity laboratory-ens 216. https://ens.sdsu.edu/dpt/research/faculty-research-interests/neuromechanics-and-neuroplasticity-lab/. accessed: 10/04/2024.

[20] Faisal Kamiran and Toon Calders. “Data preprocessing techniques for classification without discrimination”. In: *Knowledge and information systems* 33.1 (2012), pp. 1–33.

[21] Viktor Johansson. *A sensor orientation and signal preprocessing study of a personal fall detection algorithm*. 2020.

[22] Pieter Kubben, Michel Dumontier, and Andre Dekker. “Fundamentals of clinical data science”. In: (2019), p. 88.

[23] Felipe Tomazelli Lima and Vinicius MA Souza. “A Large Comparison of Normalization Methods on Time Series”. In: *Big Data Research* 34 (2023), p. 100407.

[24] Haben Yhdego, Michel Audette, and Christopher Paolini. “Fall Detection Using Self-Supervised Pre-Training Model”. In: *2022 Annual Modeling and Simulation Conference (ANNSIM)*. IEEE. 2022, pp. 361–371.

[25] I Putu Edy Suardiyana Putra and Rein Vesilo. “Window-size impact on detection rate of wearable-sensor-based fall detection using supervised machine learning”. In: *2017 IEEE life sciences conference (LSC)*. IEEE. 2017, pp. 21–26.

[26] Haziqah Shamsudin et al. “Combining oversampling and undersampling techniques for imbalanced classification: A comparative study using credit card fraudulent transaction dataset”. In: *2020 IEEE 16th international conference on control & automation (ICCA)*. IEEE. 2020, pp. 803–808.

[27] Sungkook Hong et al. “Monitoring safety behaviors of scaffolding workers using Gramian angular field convolution neural network based on IMU sensing data”. In: *Automation in Construction* 148 (2023), p. 104748.

[28] Pian Qi, Diletta Chiaro, and Francesco Piccialli. “FL-FD: Federated learning-based fall detection with multimodal data fusion”. In: *Information Fusion* 99 (2023), p. 101890.

[29] Zhiguang Wang and Tim Oates. “Imaging time-series to improve classification and imputation”. In: *arXiv preprint arXiv:1506.00327* (2015).

[30] Chao-Lung Yang et al. “Multivariate time series data transformation for convolutional neural network”. In: *2019 IEEE/SICE International Symposium on System Integration (SII)*. IEEE. 2019, pp. 188–192.

[31] Chao-Lung Yang, Zhi-Xuan Chen, and Chen-Yi Yang. “Sensor classification using convolutional neural network by encoding multivariate time series as two-dimensional colored images”. In: *Sensors* 20.1 (2019), p. 168.

[32] Navid Mohammadi Foumani et al. “Deep learning for time series classification and extrinsic regression: A current survey”. In: *arXiv preprint arXiv:2302.02515* (2023).

[33] Dongmei Han, Qigang Liu, and Weiguo Fan. “A new image classification method using CNN transfer learning and web data augmentation”. In: *Expert Systems with Applications* 95 (2018), pp. 43–56.

[34] Khalid M Hosny, Mohamed A Kassem, and Mohamed M Fouad. “Classification of skin lesions into seven classes using transfer learning with AlexNet”. In: *Journal of digital imaging* 33 (2020), pp. 1325–1334.

[35] Shui-Hua Wang et al. “Alcoholism identification based on an AlexNet transfer learning model”. In: *Frontiers in psychiatry* 10 (2019), p. 454348.

[36] Olga Russakovsky et al. “ImageNet Large Scale Visual Recognition Challenge”. In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

[37] Xiaobing Han et al. “Pre-trained alexnet architecture with pyramid pooling and supervision for high spatial resolution remote sensing image scene classification”. In: *Remote Sensing* 9.8 (2017), p. 848.

[38] Emtiaz Hussain et al. "Deep learning based binary classification for alzheimer's disease detection using brain mri images". In: *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. IEEE. 2020, pp. 1115–1120.

[39] Siyuan Lu, Zhihai Lu, and Yu-Dong Zhang. "Pathological brain detection based on AlexNet and transfer learning". In: *Journal of computational science* 30 (2019), pp. 41–47.

[40] Lokesh Borawar and Ravinder Kaur. "ResNet: Solving vanishing gradient in deep networks". In: *Proceedings of International Conference on Recent Trends in Computing: ICRTC 2022*. Springer. 2023, pp. 235–247.

[41] Larry R Medsker and LC Jain. "Recurrent neural networks". In: *Design and Applications* 5.64-67 (2001), p. 2.

[42] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[43] Lee-Ren Yeh et al. "A deep learning-based method for the diagnosis of vertebral fractures on spine MRI: retrospective training and validation of ResNet". In: *European Spine Journal* 31.8 (2022), pp. 2022–2030.

[44] Maciej Wielgosz, Andrzej Skoczeń, and Matej Mertik. "Using LSTM recurrent neural networks for monitoring the LHC superconducting magnets". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 867 (2017), pp. 40–50.

[45] Venous Roshdibenam. *Fall risk analysis using machine learning, the Timed Up and Go test, and inexpensive wearable IMU sensors*. The University of Iowa, 2021.

[46] Seung Su Jeong, Nam Ho Kim, and Yun Seop Yu. "Fall Detection System Based on Simple Threshold Method and Long Short-Term Memory: Comparison with Hidden Markov Model and Extraction of Optimal Parameters". In: *Applied Sciences* 12.21 (2022), p. 11031.

[47] Tara N Sainath et al. "Convolutional, long short-term memory, fully connected deep neural networks". In: *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. Ieee. 2015, pp. 4580–4584.

[48] David Kreuzer and Michael Munz. "Deep convolutional and LSTM networks on multi-channel time series data for gait phase recognition". In: *Sensors* 21.3 (2021), p. 789.

[49] Xiaoqun Yu, Hai Qiu, and Shuping Xiong. "A novel hybrid deep neural network to predict pre-impact fall for older people based on wearable inertial sensors". In: *Frontiers in bioengineering and biotechnology* 8 (2020), p. 63.

[50] Jun Xu, Zunwen He, and Yan Zhang. "CNN-LSTM combined network for IoT enabled fall detection applications". In: *Journal of Physics: Conference Series*. Vol. 1267. 1. IOP Publishing. 2019, p. 012044.

[51] Kei Ishida et al. "Use of 1D-CNN for input data size reduction of LSTM in Hourly Rainfall-Runoff modeling". In: *arXiv preprint arXiv:2111.04732* (2021).