Development of an Advanced Thermal Imaging-Based Human Fall Detection System

From Benchmark Dataset Collection to State-of-the-Art Model Architecture

Christopher Silver

Electrical and Computer Engineering

Lakehead University, Thunder Bay, Ontario

A thesis submitted to Lakehead University in partial fulfillment
of the requirements for the Master of Science degree
in the Electrical and Computer Engineering

Thesis Committee Members

The members listed below served on the Examining Committee for this thesis:

Supervisor:	Dr. Thangarajah Akilan Department of Software Engineering.
Committee Members:	Dr. Abdulsalam Yassine Department of Software Engineering. Dr. Yong Deng Department of Software Engineering.
Session Chair:	Dr. Yushi Zhou Department of Electrical and Computer Engineering.

Declaration of Co-Authorship / Previous Publications

I. Co-Authorship Declaration

I hereby declare that this dissertation includes material resulting from research publications conducted under the supervision of Dr. Thangarajah Akilan.

In all other parts of this dissertation, I am the primary author, having undertaken the main responsibilities, including idea generation, experimental design, data analysis, interpretation, and writing. The contributions of the co-authors in these instances were limited to proofreading and technical guidance.

I am fully aware of the Lakehead University Policy on Authorship and affirm that I have properly acknowledged the contributions of other researchers to this dissertation. Additionally, I have obtained permission from each co-author of the respective conference publications mentioned in Section II on page iii to include relevant content in this dissertation.

With these clarifications, I certify that this dissertation and the research it encompasses are my original work.

II. Declaration of Previous Publications

This thesis incorporates the content of three original research papers, which have either been previously published or awaiting acceptance in academic conferences or journals. The details are as follows:

Thesis chapter	Publication title/full citation	Status
Chapter 3	**C. Silver** and T. Akilan, "A Novel Approach for Fall Detection Using Thermal Imaging and a Stacking Ensemble of Autoencoder and 3D-CNN Models," in *2023 IEEE Canadian Conference on Electrical and Computer Engineering*, 2023, pp. 71-76.	Published
Chapter 4	**C. Silver** and T. Akilan, "TF-66: A Robust Dataset for Thermal Imaging-based Fall Detection and Eldercare," in *Journal of Engineering Applications of Artificial Intelligence*, 2024.	Submitted
Chapter 5	**C. Silver** and T. Akilan, "Real-Time Thermal Fall Detection Using Optical Flow and Attention-Enhanced Convolutional Recurrent Architectures" in *International Conference on Machine Learning ICML*, 2025.	Submitted

III. General

I declare that, to the best of my knowledge, this thesis does not infringe on any copyrights or violate proprietary rights. All ideas, techniques, quotations, or other materials derived from the work of others, whether published or unpublished, are fully acknowledged in accordance with standard referencing practices. Additionally, where copyrighted material exceeds the limits of fair dealing as defined by the Canada Copyright Act, permission has been obtained. This is a true copy of my thesis, including all final revisions as approved by my thesis committee and the Graduate Studies office. This thesis has not been submitted for a higher degree at any other university or institution.

Acknowledgements

I wish to express my sincere gratitude to Dr. Thangarajah Akilan, whose mentorship and encouragement have been a guiding force throughout my academic journey. His dedication to impactful research and his unwavering optimism provided the inspiration and focus that brought this work to life. His expertise and thoughtful guidance were critical in shaping and elevating this thesis.

My heartfelt thanks also go to the thesis committee members, Dr. Abdulsalam Yassine, Dr. Yong Deng, and Dr. Yushi Zhou, whose feedback and guidance were instrumental in successfully completing this research.

Lastly, I owe a profound debt of gratitude to my family, whose unwavering belief in me has been my source of strength through every challenge.

Dedication

This thesis is dedicated to my parents and my grandmother, whose life inspired this research. It is also dedicated to all individuals living with hemophilia, who should never be limited in their potential, and to everyone who has lost a loved one to a fall.

Abstract

Falls represent a significant risk to the elderly population, often leading to severe injuries or fatalities. Automatic fall detection systems (FDS) are critical for mitigating these risks; however, existing solutions, despite reporting accuracies in controlled environments, often fail to generalize to real-world conditions. This performance gap stems from limitations in existing datasets, overfitted models, and a lack of standardization. To address these challenges, this thesis presents a comprehensive framework for fall detection, leveraging privacy-preserving thermal imaging to develop deployable, real-world solutions.

The research is conducted in three progressive phases. The first phase explores a novel hybrid architecture that combines supervised and unsupervised learning paradigms through a stacking ensemble of 3D Convolutional Neural Networks (3D CNNs) and Autoencoders (AEs). This hybrid approach demonstrates significant performance improvements on constrained datasets, highlighting its potential in scenarios where fully supervised methods fall short. Ablation studies validate the architecture's utility while underscoring the critical need for a more robust dataset to achieve true generalizability.

In the second phase, the thesis introduces Thermal Fall 66 (TF-66), the most diverse and comprehensive thermal fall detection dataset to date. Designed to address the limitations identified in Phase 1, TF-66 encompasses varied environments, participant demographics, and fall scenarios. Accompanied by targeted subsets and flexible data generators, TF-66 serves as a benchmark for meaningful comparisons and standardized evaluations, advancing the field toward real-world applicability.

The third phase synthesizes insights from the hybrid approach and TF-66 dataset to refine a supervised 3D CNN model. Enhanced with innovative features such as optical flow integration and attention mechanisms, this model achieves state-of-the-art performance on TF-66 and the widely used Thermal Simulated Fall (TSF) dataset. By bridging the gap between lab-optimized systems and real-world demands, this final phase establishes a transformative approach to fall detection, redefining the state of fall detection research, with a focus on generalizable systems that can operate in real-time. The findings provide a clear path for developing accurate, privacy-preserving, and scalable fall detection systems, ultimately aiming to enhance the safety and save lives of seniors worldwide.

Table of Contents

List of Figures

List of Tables

List of Key Acronyms

Acronym & its Full Form	**Synopsis**
3D CNN: 3D Convolutional Neural Network	A deep learning model that uses 3D convolutional layers to capture spatial and temporal features in volumetric data, such as video sequences or 3D medical images.
ADAM: Adaptive Moment Estimation	An optimization algorithm for training deep learning models, combining the advantages of momentum and RMSProp to adapt learning rates for each parameter.
ADLs: Activities of Daily Living	Basic self-care tasks such as eating, bathing, dressing, and mobility, often used as benchmarks in fall detection systems to distinguish falls from normal activities.
AE: Autoencoder	An unsupervised neural network designed to encode input data into a lower-dimensional representation and then decode it back to the original input, minimizing reconstruction loss.
Bi-ConvLSTM: Bidirectional Convolutional Long Short-Term Memory	A deep learning model combining convolutional layers with bidirectional LSTMs to capture spatiotemporal patterns in sequential data.
Bi-LSTM: Bidirectional Long Short-Term Memory	A variation of LSTM where two LSTMs are trained, one processing the data forward and the other backward, capturing context from both directions for better predictions.

Acronym & its Full Form	**Synopsis**
BCE: Binary Cross-Entropy (Log Loss)	A loss function for binary classification problems, minimizing the difference between predicted probabilities and actual binary label.
CAE: 3D Convolutional Autoencoder	A type of convolutional autoencoder that processes 3D data, such as video frames, to learn spatiotemporal features through 3D convolutional layers.
CNN: Convolutional Neural Network	A deep learning model designed for processing structured data such as images, leveraging convolutional layers to detect spatial features and patterns at different abstraction levels automatically.
CNN-LSTM: IConvolutional Neural Network - Long Short-Term Memory	A hybrid deep learning model that combines CNN layers for spatial feature extraction with LSTM layers to capture temporal dependencies in sequential data.
ConvAEs: Convolutional Autoencoders	A type of autoencoder that uses convolutional layers to learn spatial hierarchies in image data during encoding and decoding processes.
ConvLSTM-AE: Convolutional LSTM Autoencoder	An advanced architecture that combines convolutional and LSTM layers to capture spatial and temporal features in sequential data.
ConvLSTM: Convolutional Long Short-Term Memory	A deep learning model that integrates convolutional operations within LSTM layers to process spatiotemporal data in tasks such as video analysis and sequential prediction.

Acronym & its Full Form	Synopsis
CTS-EVK: Calumino Thermal Sensor Evaluation Kit)	A compact thermal sensor capturing raw thermal images at 35×15 pixels, upsampled to 140×60 pixels. It features a built-in human presence detector and connects via USB or MQTT to Calumino software for recording and playback, supporting user-friendly and efficient data collection.
DAEs: Denoising Autoencoders	A type of autoencoder designed to reconstruct input data from partially corrupted versions, enabling the model to learn robust feature representations by minimizing reconstruction errors.
DL: Deep Learning	A subset of artificial intelligence that utilizes neural networks with multiple layers to analyze data and extract meaningful patterns, often applied in tasks such as image recognition, natural language processing, and time-series analysis.
FPR: False Positive Rate	A performance metric in binary classification that measures the proportion of negative samples incorrectly classified as positive. It is calculated as False Positives divided by the sum of False Positives and True Negatives.
FDS: Fall Detection Solutions	Systems designed to detect falls using various sensor modalities and machine learning approaches for timely alerts and assistance.
GFLOPS: Giga Floating Point Operations Per Second	A measure of computational performance, indicating the number of billions of floating-point operations a system performs per second.
GPU: Graphics Processing Unit	A specialized processor designed for parallel computation, widely used in deep learning and image processing tasks.
GRU: Gated Recurrent Unit	A simplified version of LSTM that uses gating mechanisms to manage the flow of information without separate memory cells, making it computationally more efficient.

Acronym & its Full Form	Synopsis
IR: Infrared	A sensor modality that detects thermal radiation for applications such as privacy-preserving fall detection systems.
LSTM: Long Short-Term Memory	An advanced type of RNN capable of learning long-term dependencies through specialized memory cells that manage the flow of information.
ML: Machine Learning	A subset of artificial intelligence that trains models to identify patterns and make predictions from data.
MLP: Multi-Layer Perceptron	A fully connected neural network architecture commonly used in supervised learning tasks.
mmWave: Millimeter Wave	A technology that uses electromagnetic waves with wavelengths in the millimeter range (30 GHz to 300 GHz). It is widely used in applications such as radar, imaging, and communication systems, including fall detection.
MUVIM: Multi-View Motion Dataset	A thermal dataset with 400 falls from 30 participants using high-resolution cameras. Limitations include privacy concerns, reliance on costly multi-sensor setups, absence of elderly participants, and issues with FLIR camera reliability and data loss.
PIR Sensors: Pyroelectric Infrared Sensors	Sensors that detect motion by measuring changes in infrared radiation levels emitted by objects in their field of view. They are often used in low-cost and privacy-preserving fall detection systems.
PSIT: Per Sample Inference Time	A metric for evaluating the computational complexity and feasibility of real-time performance for deep learning models.
RF: Radio Frequency	Technology using electromagnetic waves for communication and sensing applications, often employed in non-invasive fall detection.

Acronym & its Full Form	Synopsis
RGB: Red Green Blue	A color model used for representing visual data, where images are composed of three color channels (red, green, and blue). RGB data is often used in vision-based systems like fall detection.
RNNs: Recurrent Neural Networks	A type of neural network designed for sequential data, where connections between nodes form directed cycles, enabling persistence of information over time.
ROC-AUC: Receiver Operating Characteristic - Area Under the Curve	A performance metric for classification models, measuring the trade-off between true positive rate and false positive rate across different thresholds.
SRAE: Spatiotemporal Residual Autoencoder	A deep learning model that combines residual connections with spatiotemporal feature extraction to improve the learning of spatial and temporal data patterns, often used in video and sequential data analysis.
TF-66: Thermal Fall 66 (dataset)	A public dataset created in this work, specifically designed for fall detection using thermal imaging, addressing limitations in existing datasets by incorporating diverse scenarios and physiotherapist-guided simulated falls.
TSF: Thermal Simulated Fall Dataset	A benchmark dataset for thermal fall detection, featuring 35 fall and 9 non-fall videos with a 480x640 grayscale resolution. Addresses class imbalance but exhibits limitations like a small dataset size.
UWB: Ultra Wide Band	A wireless communication technology that operates across a wide frequency spectrum, offering precise location tracking and low-power operation, suitable for indoor positioning and activity recognition applications.

Chapter 1

Introduction

1.1 Thesis Overview

The field of automatic fall detection has faced stagnation in recent years. While some research claims accuracies exceeding 99% in controlled lab environments, these systems often fail to meet the expectations of industry partners who find them too unreliable for real-world deployment. At the core of this issue lies a lack of generalizability. High-performing fall detection models are frequently overfitted to small, non-diverse datasets, typically consisting of a limited number of participants and environments. Consequently, these systems may perform well in idealized lab conditions but lack the robustness to handle the variability inherent in real-world scenarios.

The absence of standardized datasets further exacerbates this problem, hindering meaningful comparisons between models and interpretations of performance metrics. This lack of standardization has fragmented the field, with researchers focusing on diverse modalities—such as wearable sensors, RGB cameras, radar solutions, and depth cameras—and varying methodological approaches. While some researchers focus exclusively on supervised models, others explore unsupervised or hybrid techniques. This diversity of research is valuable for innovation but highlights a critical disconnect between lab-developed fall detection systems and the requirements of real-world implementation. Bridging this gap necessitates convergence on a practical fall detection modality, the creation of a comprehensive and representative dataset, and the optimization of

models to perform reliably under real-world conditions. These advancements would enable researchers to compare architectures fairly and develop systems that reflect true performance rather than idealized lab outcomes. The absence of robust datasets and benchmarks continues to hinder the identification of the most effective machine learning approaches for fall detection.

This thesis seeks to realign fall detection research with real-world needs by developing models explicitly designed for practical deployment, ultimately aiming to save lives. Through a pragmatic and systematic approach, this work builds a robust knowledge base and advances the development of a state-of-the-art fall detection system, evaluated on the most comprehensive dataset in the field.

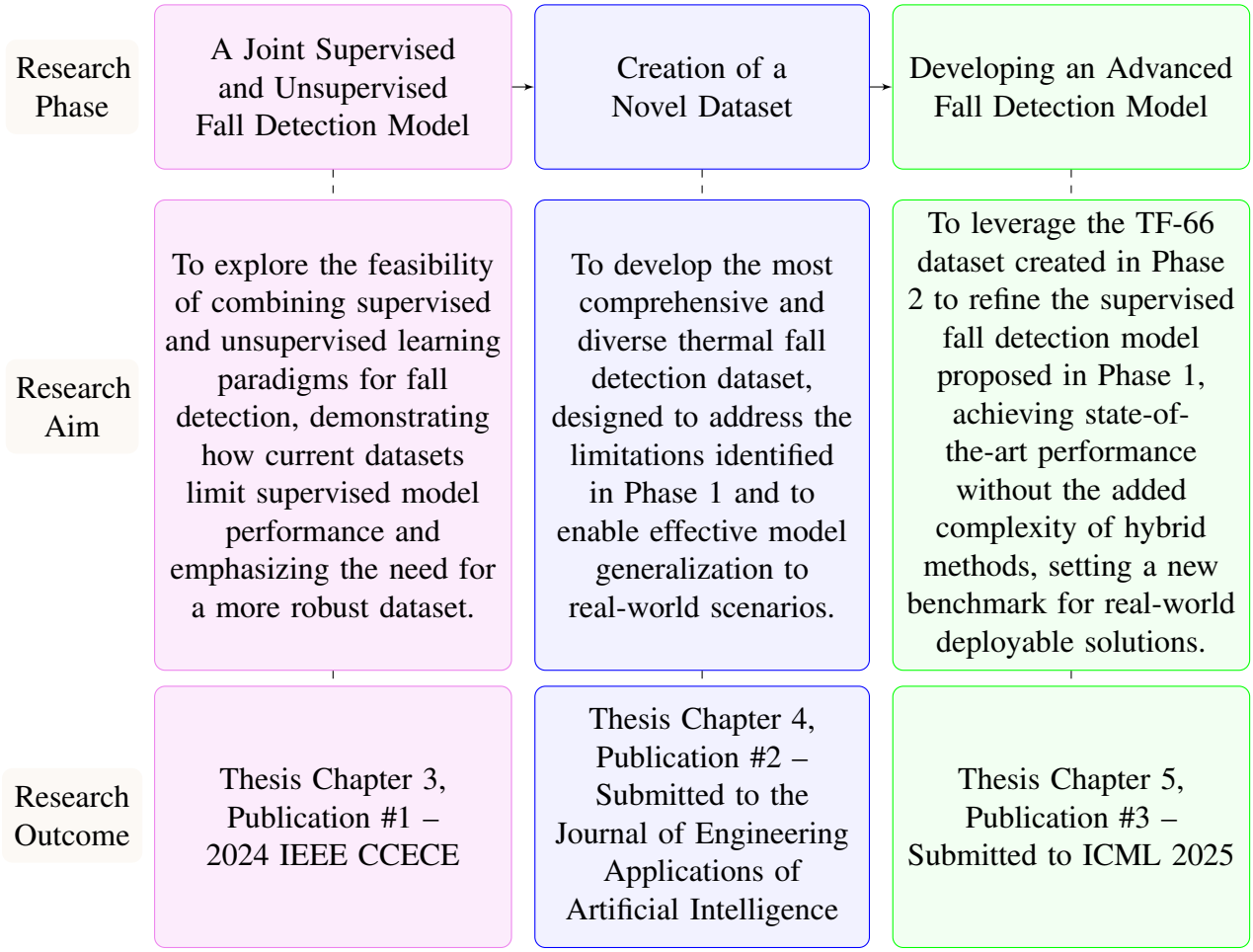


Figure 1.1: Diagram of the thesis road map.

Note: It outlines three progressive stages, each focused on acquiring the foundational knowledge and expertise necessary to achieve the thesis's ultimate goal.

To ensure this research addresses the needs of end-users and industry stakeholders, it incorporates insights from literature trends and feedback from seniors—the primary users of these systems.

The first phase of this thesis explores the feasibility of a novel hybrid approach that combines supervised and unsupervised learning for fall detection—an approach not previously examined in the field (cf. Chapter 3). This phase highlights the limitations of existing datasets, which often lack the diversity and robustness required for supervised models to generalize effectively. Initial findings demonstrate the potential of hybrid approaches to bridge these gaps, while underscoring the need for a more comprehensive dataset to fully realize the promise of fall detection systems.

In the second phase, the limitations identified in Phase 1 are addressed by introducing TF-66, the most robust and diverse fall detection dataset to date. Capturing a wide range of environments, demographics, and fall scenarios, TF-66 surpasses existing datasets in scope and quality. This dataset establishes a critical benchmark for future research and enables standalone supervised models to achieve unprecedented levels of performance and generalizability (cf. Chapter 4).

Finally, the third phase synthesizes the findings of the previous two phases by refining the supervised 3D CNN model introduced as part of the hybrid architecture in Phase 1. With the robust TF-66 dataset, the refined model achieves state-of-the-art performance, eliminating the need for the added complexity of hybrid approaches. This phase demonstrates that the diversity and quality of TF-66 allow standalone supervised models to meet real-world demands, setting a new standard for fall detection systems. This final phase fulfills the thesis's primary objective: to realign fall detection research with real-world applicability and safety, paving the way for solutions capable of saving lives globally (cf. Chapter 5).

1.2 Motivation

Falls, as defined by the World Health Organization (WHO), involve "inadvertently coming to rest on the ground or lower level, excluding intentional changes in position." Unattended falls are a leading cause of both fatal and non-fatal injuries among seniors, posing a growing concern as the global population ages [3–5]. This issue is becoming increasingly critical as the global demo-

graphic shifts toward an aging population. In Canada, the proportion of individuals aged 65 or older rose from 12% in 2005 to 20% in 2020, with this trend expected to continue [6]. Worldwide, the number of seniors is projected to nearly double between 2020 and 2050 [7, 8]. This growth is driven by the aging of the baby boomer generation, declining birth rates [9], and advances in healthcare, which have extended life expectancy. As a result, the number of elderly citizens will soon equal that of younger adults, exacerbating the current shortage of caretakers and senior living facilities [7, 10, 11]. With this increasing aging population, fall detection has become a vital tool in enabling seniors to live safely and independently. In response to this worldwide demographic shift, the United Nations has called for the development of environments that enable and support seniors [12]. Automatic fall detection devices represent a critical technology in addressing these needs by offering real-time detection and response, thereby preventing prolonged medical emergencies.

1.2.1 Impact of Falls on Seniors

Falls are the second leading cause of accidental death among seniors, contributing to nearly 700,000 deaths globally each year [8, 13–15]. They account for approximately half of all accidental injuries in seniors [16] and 40% of all injury-related fatalities [17]. Alarmingly, nearly 5% of individuals hospitalized due to a fall do not survive their hospital stay, and approximately 20% of those who experience a fall will die within a year of the incident [18]. The risk and severity of falls increase with age, as balance and bone strength naturally decline over time. In older women, bone demineralization occurs at a rate of 3-6% per year, while older men experience a loss of 0.5-2% annually [12]. Even when falls are not fatal, they can have serious consequences. Approximately 20% of falls result in a serious head injury or broken bones, with falls being the leading cause of fractures in seniors [1, 13, 19]. Hip fractures, for instance, are predominantly caused by falls (98%) and are associated with a 1 in 4 chance of long-term institutionalization and a 1 in 5 chance of death [20]. Beyond physical injuries, falls can profoundly impact the mental health of seniors. Many seniors experience a fear of falling again, which can lead to reduced mobility and physical deterioration, further increasing the risk of future falls [18, 21]. This fear-induced reduction in movement can

perpetuate a cycle of decreased balance, reduced confidence, and increased fall risk, ultimately limiting independence and quality of life [20, 22, 23].

1.2.2 The Need for Automatic Fall Detection

An automatic fall detection device is crucial for seniors living alone, primarily due to the risk of a "long lie"—a situation where a senior falls, is unable to get up, and remains in the prone position for an extended period. Approximately 50% of seniors who recover from a long lie die within six months of the event [7, 18, 24]. Long lies can cause severe complications, such as dehydration, rhabdomyolysis, sores, and internal bleeding, often leading to death [18]. Even when the senior survives, long lies can significantly reduce their independence, mobility, and quality of life [21, 25].

Seniors who experience a long lie after falling do not necessarily lose consciousness; many are unable to get up independently. Approximately 30% of seniors over the age of 90 experience a long lie following a fall [26]. Timely detection of falls is critical, as it significantly improves recovery outcomes by enabling faster medical intervention [7, 11, 27]. Prompt medical attention after a fall can reduce the risk of death by 80% and decrease the likelihood of long-term hospitalization by 26% [28]. Fall-related injuries impose significant financial burdens. In Canada, one in three seniors experiences a fall annually, resulting in direct costs exceeding $2 billion CAD per year [29]. In the United States, the estimated cost of falls reached $55 billion USD in 2020 [30]. Health care costs rise by 29% for seniors who experience a single fall and by 79% for those who fall multiple times in a year [18].

As the population ages, the availability of long-term care facilities perpetually becomes more limited. Seniors often prefer to live independently, but concerns from family members about safety, health, and fall risks may pressure them to move into care homes [13, 31, 32]. Automatic fall detection systems can provide peace of mind for both seniors and their families, allowing seniors to remain in their homes safely. Even in residential care settings, fall detection can be beneficial, as seniors often fail to report falls due to embarrassment, leading to unaddressed injuries [18]. Automatic fall detection systems are not just for seniors. Individuals of all ages who suffer from cardiovascular diseases, neurodegenerative disorders, diabetes, or progressive conditions like

Parkinson's are also at risk of falls, with over 10% of patients with Parkinson's falling more than once a week [33]. Additionally, among the 70 million wheelchair users worldwide, 60% experience falls, and 80% of these individuals require assistance after falling [15, 17, 20, 23]. Accidents often occur in industrial fields in workers or all ages, with falling being a top safety concern. FDS could be installed to mitigate these concerns in the industrial field as well [34].

1.2.3 Challenges and Research Gaps

Fall detection presents a unique challenge as it requires the integration of spatial and temporal features to accurately identify fall events. This spatiotemporal complexity demands advanced deep learning models that traditional frameworks and data generators cannot adequately support. Unlike static image-based tasks, fall detection relies on 3D data, such as sequences of consecutive video frames, which significantly increases computational requirements. Real-time deployment of these systems further compounds the challenge, particularly on resource-constrained devices like edge platforms.

Existing datasets add to these difficulties. Most datasets are small, lack diversity, and are tailored to controlled environments, limiting the ability of models to generalize to real-world scenarios. The absence of standardization in dataset design and evaluation methods fragments research efforts and hinders meaningful comparisons across models. Additionally, high false alarm rates and poor real-time performance remain critical barriers, undermining user trust and system adoption. Sensor-specific challenges also persist: wearable devices face adherence and usability issues, ambient sensors are often restricted by coverage and calibration requirements, and vision-based solutions are hindered by occlusion, lighting conditions, and privacy concerns.

Despite these advancements, fall detection systems have yet to achieve widespread commercial adoption. The ultimate goal remains to deliver solutions that can operate reliably in diverse real-world environments and save lives. This thesis addresses these challenges by bridging the gap between academic research and practical deployment, with an emphasis on real-world applicability and user preferences. To determine the most effective path forward for fall detection systems,

existing methods must be rigorously examined, compared, and contrasted, paving the way for a solution that meets the needs of both users and industry stakeholders.

1.3 Taxonomy of FDS

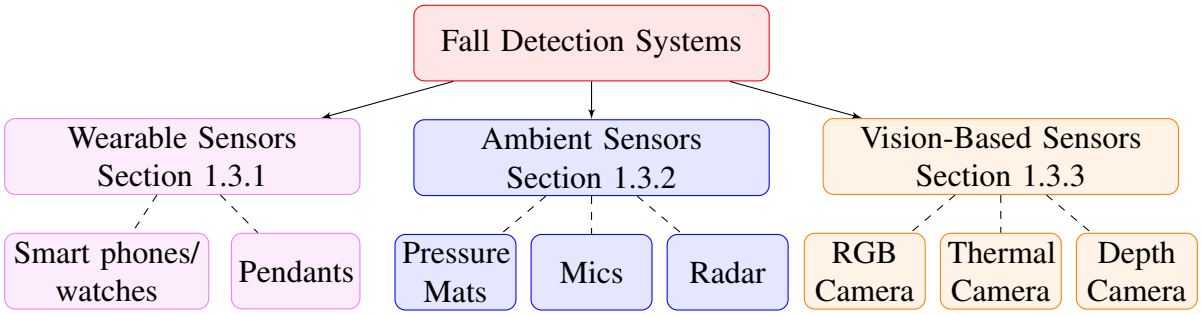


Figure 1.2: A taxonomy of fall detection systems.
Note: This taxonomy organizes fall detection systems into three main categories—wearable, ambient, and vision sensors—highlighting the range of hardware implementations used within each category.

Current fall detection systems can be grouped into three main categories: wearable sensors, ambient-based sensors, and vision-based solutions. Each of these approaches has its merits but suffers from significant limitations, making them insufficient for real-world deployment, particularly in senior care environments. A summary of a taxonomy of fall detection systems (FDS) can be seen in Figure 1.2.

1.3.1 Wearable Sensors

Wearable sensors are the most common type of fall detection system and typically rely on accelerometers and gyroscopes to monitor acceleration and rotational motion of the user wearing the device. These sensors are often embedded in devices such as phones, watches, or pendants worn by the user. Examples of some common wearable sensor based solutions can be seen in Figure 1.3. Upon detecting motion characteristic of a fall, these devices can automatically trigger an alert for help. While wearable solutions have been successful in reducing risks associated with long lies when they function correctly [35], they face several critical shortcomings that undermine their ef-

fectiveness. Although most seniors are open to the idea of using fall detection devices, only about 10% actually adopt them [35, 36]. One major reason for this gap is that, despite having automatic fall detection capabilities, many devices require the user to manually press a button to call for help if the automatic system fails. Studies have shown that seniors may panic after falling and forget to activate the help button, instead resorting to calling for help on a phone if they can reach it [13]. Even more concerning, in 80% of cases where seniors fell and were unable to get up, they failed to use their wearable fall detection devices to call for help [18, 26], highlighting the critical need for reliable automatic detection.

Wearable devices also present physical usability challenges. Many seniors report difficulty pressing the help button due to its intentional resistance, designed to prevent accidental activations. However, false alarms remain common, leading to frustration and non-compliance, with some users eventually discontinuing use altogether [18, 37, 38]. False alarms may occur even during cautious activities, such as unintentional bumps, and are especially prevalent in systems that do not adequately account for post-fall posture [15, 39, 40]. Another significant limitation is the placement of sensors on the wrist, such as in smartwatches. Because of their placement, wrist-based devices exhibit greater fluctuation in measurements compared to sensors worn closer to the body's center of mass, such as on the pelvis or a belt buckle [41]. Commodity smartwatches

Smartphone for fall detection.

Smartwatch for fall detection.

Fall detection pendant.

Figure 1.3: Visual examples of wearable fall detection solutions.
Note: These wearable fall detection solutions—smartphone, smartwatch, and pendant—leverage embedded accelerometers and gyroscopes to detect motion patterns indicative of falls.

also produce noisier data than specialized equipment, and their constrained computational power further limits real-time fall detection capabilities. These systems often fail to classify all falls and generate excessive false positives, making them impractical for real-world use [41]. In addition to usability concerns, wearable devices require regular charging or battery replacements, and users must remember to wear them consistently [7, 8, 40, 42]. Many seniors do not wear these devices during sleep or when getting up at night, rendering them ineffective during these critical moments [38, 43–46]. Silent failures, where users are unaware that their device is no longer functional, are another concerning issue that can render fall detection systems useless [27]. Discomfort and social stigma also deter seniors from using these devices, with many expressing embarrassment over false alarms in public or reluctance to wear a device that symbolizes a loss of independence [13, 18, 39].

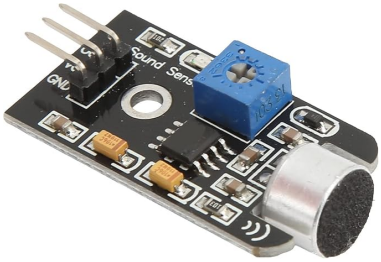
Studies have consistently shown that adherence to wearing these devices is low. In one 2015 study, participants wore a wearable fall detection pendant for up to four months. During this period, only one fall was correctly detected, while 84 false alarms occurred, and 11 falls went undetected. Of these undetected falls, 8 occurred because participants were not wearing the device, despite instructions to do so [18]. This suggests that adherence issues, combined with technical limitations, contribute to significant lapses in fall detection. Moreover, the performance claims made by manufacturers of wearable devices often fail to translate into real-world results. For instance, in the aforementioned 2015 study [18], the tested device, trained on data from 59 volunteers, achieved 94% fall detection accuracy in lab conditions but detected only 25% of falls in real-world settings. This discrepancy underscores the challenges of transitioning from controlled environments to practical applications. Additionally, regulatory oversight is limited; in the United States, the Food and Drug Administration (FDA) permits manufacturers to self-classify their devices, potentially bypassing rigorous evaluations. The device used in the 2015 study avoided FDA approval entirely, raising questions about the reliability of commercial solutions [18, 47]. Even FDA-approved devices, such as the AutoAlert fall detection pendant, have shown limitations. For example, a 2020 study reported that while the device detected 82.5% of falls in young, healthy adults simulating falls, forward falls were detected only 53.3% of the time [35].

Finally, hardware issues, such as the need for manual calibration of accelerometers and gyroscopes, further complicate the use of wearable sensors [40]. Although newer devices like smartwatches and smartphones are gaining popularity, their placement away from the body's center of mass significantly compromises accuracy, making them less reliable for fall detection [12, 13, 33]. Overall, wearable sensors face numerous challenges, including adherence issues, high false positive rates, and poor performance in real-world conditions, limiting their utility as a reliable fall detection solution for seniors.

1.3.2 Ambient Sensors

Ambient sensors, also known as environmental sensors, provide a non-intrusive alternative to wearable devices by being installed in the user's living space rather than worn on the body. These systems include pressure mats, motion detectors, microphones, and radar technologies [48, 49]. Examples of some common wearable sensor based solutions can be seen in Figure 1.4. A major advantage of ambient sensors is that they eliminate adherence issues; users do not need to remember to wear or charge a device, allowing for seamless integration into daily life.

A pressure mat, which sends alerts when the user falls off of it.



A microphone to detect loud noises, indicating a fall.

A router to collect WiFi information for fall detection.

Figure 1.4: Visual examples of ambient fall detection solutions.
Note: These ambient fall detection solutions include a pressure mat for detecting weight shifts, a microphone for identifying loud noises associated with falls, and a router for analyzing changes in WiFi signal patterns. These devices offer non-invasive, environmental-based fall detection methods.

However, non-radar ambient sensors face significant challenges. Microphone-based systems, for example, are highly sensitive to background noise, which can lead to high false alarm rates, particularly in noisy environments [50, 51]. This issue is exacerbated in retirement facilities, where shared living spaces often produce significant ambient noise. Pressure mats and motion detectors, while useful, are typically confined to specific locations, limiting their coverage and requiring careful placement for effective monitoring [19]. Focus group studies indicate that these systems are often perceived as invasive, with participants expressing discomfort about being constantly monitored [13]. This discomfort, combined with the high rate of false alarms, suggests that ambient sensors may not be widely accepted as a solution among seniors.

Radar-based systems have emerged as promising solutions for fall detection, offering immunity to external factors such as noise, lighting conditions, and smoke [9]. Utilizing reflected Radio Frequency (RF) signals, these systems can detect motion and "see" through walls, providing privacy-preserving, non-contact monitoring in diverse conditions [9]. Common radar technologies include Ultra-Wideband (UWB), Millimeter-Wave (mmWave), and Frequency Modulated Continuous Wave (FMCW) radars, each with unique characteristics.

Ultra-Wideband Radar leverages body-reflected radio waves to measure distances and detect motion through walls [9, 16]. While Millimeter-Wave Radar offers higher spatial and velocity resolution than UWB, enabling the detection of micro-motions, such as breathing [52]. Frequency Modulated Continuous Wave Radar combines the capabilities of UWB and mmWave, tracking both distance and velocity for multiple targets simultaneously [9]. WiFi-Based Systems detect motion by analyzing signal distortions caused by human activity. These systems are low-cost and leverage existing infrastructure, making them an accessible alternative to radar [50]. Despite their advantages, radar-based and WiFi systems face critical limitations that hinder their effectiveness as automatic fall detection modalities. High false alarm rates and difficulties in identifying specific individuals in shared environments, such as retirement facilities, remain significant obstacles. Some radar systems require line-of-sight or extensive calibration to mitigate environmental interference, while their reliance on complex signal processing can make them computationally demanding and impractical for real-world deployment. Similarly, WiFi signals are highly sensitive to environmen-

tal noise, furniture placement, and crowded frequency bands, further reducing their reliability in real-world scenarios. These limitations, along with issues of dataset generalizability and ambiguity in multi-occupant settings, render these technologies sub-optimal for deployment in unstructured environments, particularly where precise fall detection is essential.

1.3.3 Vision-Based Solutions

Vision-based solutions are passive systems that use cameras to monitor a senior's living space. These systems continuously capture video feeds, which are processed by deep learning models to detect falls. Examples of some common vision-based sensor solutions can be seen in Figure 1.5. One key advantage of vision-based systems is that they eliminate compliance issues, as users are not required to wear or interact with any devices for the system to function [11, 53]. Additionally, these systems are often hardwired, removing concerns about battery life or charging [15]. However, vision-based solutions are generally limited to the home due to their stationary nature, which many potential users find acceptable given that falls most often occur in familiar environments [22, 54].

RGB cameras, which capture red, green, and blue light, are a popular choice for vision-based fall detection systems due to their ability to provide clear and detailed images under well-lit conditions. However, they face significant limitations in real-world applications. One of the primary

An RGB camera. Captures standard visual images for analyzing falls.

A thermal sensor. Detects heat signatures to monitor fall events.

A depth sensor. Measures the distance between objects for fall detection.

Figure 1.5: Visual examples of vision-based fall detection solutions.
Note: Vision-based fall detection solutions leverage these sensors to extract critical features: RGB cameras for capturing visual imagery, thermal sensors for privacy-preserving heat detection, and depth sensors for analyzing spatial relationships and object positioning. Each sensor plays a unique role in improving fall detection accuracy across different environments.

challenges is their poor performance in low-light or nighttime environments, where the absence of sufficient visible light severely impacts their reliability [39, 55]. While falls are more common during the day, nighttime falls remain a critical concern that cannot be overlooked, as RGB cameras struggle to provide accurate detection in these scenarios [8]. Privacy concerns further limit the adoption of RGB-based systems. Many seniors are uncomfortable with constant video surveillance in their homes, perceiving it as an intrusion on their personal space [13, 39]. This discomfort is well-documented in both focus groups and existing literature [8, 15, 18, 39, 55], highlighting a significant barrier to the practical deployment of such systems.

Occlusion presents another challenge, as furniture or other objects may obstruct the camera's view, reducing detection accuracy [18, 55]. These systems also struggle with generalization across diverse environments. Each home introduces unique variations in camera angles, lighting conditions, and furniture layouts, making it difficult to train machine-learning models capable of performing reliably in all settings. Even if systems were customized for individual homes, they would still need to account for the wide range of possible fall scenarios and lighting conditions, a task that is often impractical [8, 50]. These limitations collectively undermine the reliability of RGB-based fall detection systems in real-world applications. While they offer certain advantages over wearable devices, their dependency on visible light, privacy concerns, sensitivity to occlusion, and lack of generalizability across environments pose significant challenges that must be addressed to make them viable for real-world deployment.

Depth Cameras: Depth cameras, such as those popularized by the Microsoft Kinect sensor, capture depth information using methods like infrared (IR) structured light systems, Time of Flight, or stereo vision [56]. Unlike RGB cameras, depth cameras are relatively independent of visible light, enabling them to function effectively in low-light conditions [56]. However, these sensors have notable limitations. Time of Flight sensors are susceptible to sunlight interference, while stereo vision systems perform poorly in low-texture areas [56]. Additionally, depth cameras are range-limited, restricting their applicability in larger spaces. Despite these challenges, depth cameras

present a viable alternative to RGB cameras, particularly in environments with varying lighting conditions.

Thermal Cameras: Thermal cameras detect temperature differences between objects and their surroundings, relying on thermal radiation rather than visible light. This capability makes them uniquely effective in low-light or no-light environments, offering significant advantages for fall detection in scenarios where lighting conditions are unpredictable [56]. Unlike RGB and depth cameras, thermal cameras are not affected by lighting variations or occlusion caused by reflective surfaces. However, thermal cameras have historically been constrained by their lower resolution compared to RGB cameras and their relatively high cost, although prices have been decreasing over time [56]. These limitations are increasingly being mitigated with advancements in thermal sensor technology, broadening their potential for real-world deployment.

In summary, the existing fall detection solutions continue to face significant challenges in reliability, user compliance, and practicality. RGB and depth cameras are limited by lighting conditions, environmental interference, and range, while thermal cameras, despite their advantages, have been historically constrained by resolution and cost. These limitations highlight the critical need for a more comprehensive and effective solution that addresses the shortcomings of existing wearable and vision-based approaches, paving the way for robust, user-centered, and privacy-preserving fall detection systems.

1.4 Technical Approach

This thesis advances fall detection research by leveraging thermal sensors in conjunction with deep learning models to develop practical and deployable solutions for automatic fall detection. The research is structured into three distinct phases, each building upon the insights and outcomes of the previous phase:

- **Phase 1: Exploring Hybrid Architectures:** The first phase investigates the feasibility of a novel hybrid architecture combining supervised and unsupervised learning paradigms via

a stacking meta-model. While the hybrid approach is not included in the final system, this exploration provided critical insights into addressing challenges associated with small and non-representative datasets. These findings directly informed the development of the TF-66 dataset and the brute-force data generation approach, which proved vital for achieving robust supervised learning in Chapter 5. This phase also establishes a methodology that future researchers can adopt for constrained datasets (cf. Chapter 3).

- **Phase 2: Developing the TF-66 Dataset:** The second phase addresses limitations of existing fall detection research by introducing TF-66, the most robust, diverse, and privacy-preserving thermal fall detection dataset to date. TF-66 not only provides a comprehensive benchmark but also includes targeted subsets and flexible data generators to enable specialized training for diverse deployment scenarios. This dataset forms the backbone of the experimental evaluations conducted in the subsequent phase (cf. Chapter 4).

- **Phase 3: Refining and Evaluating a Supervised Model:** The third phase builds upon the insights from Phase 1 and leverages the TF-66 dataset developed in Phase 2. This phase focuses on optimizing the supervised 3D CNN architecture and integrating advanced techniques such as attention mechanisms, recurrent layers, and optical flow. The refined models are rigorously tested on both TF-66 and TSF, achieving state-of-the-art performance. This phase demonstrates the feasibility of a streamlined supervised approach, providing a practical and deployable solution tailored to real-world conditions while setting new benchmarks in thermal fall detection (cf. Chapter 5).

This structured approach ensures that the research not only addresses the limitations of existing systems but also lays a robust foundation for future advancements. By emphasizing real-world applicability and scalability, this work bridges the gap between academic research and practical deployment, paving the way for life-saving fall detection solutions.

The next section introduces foundational concepts in machine learning, computer vision, and deep learning, providing the necessary context for the solutions developed in this thesis.

1.5 Overview of Machine Learning and Computer Vision

Note To Readers: This thesis addresses numerous interconnected topics related to machine learning, computer vision, and deep learning. Due to space constraints, each topic is introduced with a high-level overview specific to the domain of fall detection rather than an in-depth explanation of the concept as a whole. Readers seeking a more comprehensive understanding of these foundational concepts are encouraged to refer to the following resources:

- **Stanford University's CS231n: Convolutional Neural Networks for Visual Recognition**: This course provides detailed explanations of machine learning fundamentals, computer vision basics, video-based computer vision, and deep learning models such as CNNs, RNNs, and autoencoders.
- **DeepLearning.AI**: An online platform offering structured courses on machine learning, deep learning, and computer vision, including practical applications and detailed theoretical insights.

The summaries provided in this section are intended to offer the necessary context for understanding the work presented in this thesis. Readers are encouraged to explore the recommended resources for a deeper dive into the topics discussed.

1.5.1 Machine Learning Types

Machine learning (ML) involves designing systems that learn patterns from data to solve problems or extract meaningful insights across various domains. The most common types of ML tasks include:

- **Classification**: Predicts discrete labels or categories based on input data. For example, determining whether an email is spam or not.
- **Regression**: Predicts continuous values, such as forecasting house prices based on features like size, location, and number of rooms.

- **Clustering**: Groups data points into clusters based on similarity, as seen in applications like customer segmentation and topic modeling.
- **Association**: Identifies relationships between variables in datasets, such as discovering items frequently purchased together in market basket analysis.

ML is applied across diverse fields, with applications often categorized based on the type of data being analyzed:

- **Computer Vision**: Focuses on understanding and interpreting visual data, such as images and videos, for tasks like object detection, facial recognition, and autonomous driving.
- **Natural Language Processing (NLP)**: Deals with understanding and generating human language in text or speech form, enabling applications like sentiment analysis and language translation.
- **Speech Recognition and Processing**: Involves interpreting spoken language, with uses in voice assistants, automated transcription, and accessibility tools.

Thermal fall detection falls under the domain of **classification using computer vision**, as it involves processing and interpreting thermal images to identify falls. Consequently, a deeper exploration of computer vision concepts is necessary.

1.5.2 Computer Vision Tasks

Computer vision tasks are designed to recognize, locate, and analyze objects or patterns in visual data. The following tasks are fundamental to computer vision:

- **Image Classification**: Categorizes an entire image into predefined classes, such as determining whether an image contains a cat or a dog.
- **Object Detection**: Identifies objects within an image and localizes them using bounding boxes, as seen in applications like pedestrian detection.

While many computer vision tasks focus on analyzing static images, fall detection is inherently a **temporal problem**—falls occur over time and often require contextual information from preceding and subsequent frames to be accurately detected. This makes video-based computer vision a more appropriate approach for fall detection.

1.5.3 Video-based Computer Vision

Video-based computer vision extends traditional computer vision to analyze temporal relationships in sequences of frames. The key tasks include:

- **Action Recognition**: Identifies actions in a video clip, like running, jumping, or falling.
- **Object Tracking**: Tracks objects across multiple frames in a video, crucial for applications like surveillance and sports analytics.
- **Video Classification**: Assigns a label to an entire video sequence based on its content, such as classifying a video as a soccer match or a cooking tutorial.
- **Anomaly Detection in Videos**: Detects rare or unexpected events in video streams, such as identifying falls or security breaches.

Fall detection is closely aligned with **video classification** and **anomaly detection**, as it requires understanding temporal sequences to identify abnormal patterns associated with falls.

1.5.4 Video Classification

Video classification involves assigning a label to an entire video sequence by analyzing both spatial and temporal information. Unlike static image classification, video classification models process sequences of frames to capture motion patterns and temporal relationships. The process typically includes preprocessing steps like frame extraction, resizing, and normalization, followed by feature extraction to identify spatial elements within frames. Temporal modeling captures motion and activity across consecutive frames, enabling the classification of dynamic scenes or events.

Recent advancements in deep learning have significantly improved video classification through architectures that combine spatial and temporal analysis. Techniques like frame sampling, optical flow computation, and transfer learning enhance model performance, making video classification effective in applications like activity recognition, surveillance, and sports analytics.

1.5.5 Anomaly Detection in Videos

Anomaly detection in videos identifies unusual or unexpected events that deviate from normal patterns. For fall detection, this involves recognizing rare behaviors, such as a person falling, amidst typical activities. The process begins with feature extraction to capture spatial and temporal cues, followed by comparing these features against models of typical behavior. DL models, particularly those leveraging spatiotemporal dynamics, excel at identifying subtle deviations over time, enabling effective anomaly detection in complex environments. Transfer learning has become a crucial tool in anomaly detection, allowing models to adapt pre-trained knowledge to specific contexts. This approach reduces training time and improves performance, particularly in scenarios where labeled anomalous data is limited. Applications range from fall detection in healthcare to security monitoring and industrial equipment analysis.

1.6 Deep Learning

DL is a subset of machine learning that uses neural networks with multiple layers to model complex patterns in large datasets. Inspired by the structure of the human brain, these networks transform input data into increasingly abstract representations, enabling models to automatically learn features directly from raw data without manual feature engineering [15, 28, 30]. This adaptability makes DL particularly effective for fall detection, as it can leverage the nuanced patterns present in thermal data [57].

One of DL's key strengths is its ability to generalize patterns from data, even when trends may not be apparent to human designers [33, 58]. However, its performance depends heavily on the availability of large, diverse datasets. Without sufficient data, models risk overfitting to the train-

ing set, leading to poor generalization in real-world scenarios [58, 59]. For instance, a fall detection model trained on thermal data from a limited number of environments may fail to perform well in new settings with different furniture layouts or lighting conditions [33, 60]. This underscores the importance of creating comprehensive datasets that reflect the range of conditions expected during deployment. DL models are trained by minimizing a loss function that quantifies the error between predictions made about the data and the ground truth labels in the training data. Through back-propagation and gradient descent, model parameters are iteratively updated to reduce this error. To prevent overfitting, datasets are typically split into training and validation subsets. The validation set, containing unseen data, is used to evaluate the model's ability to generalize before deployment. However, even strong performance on validation data does not guarantee success in real-world applications unless the dataset sufficiently captures real-world variability [33]. With comprehensive datasets, DL models have demonstrated exceptional performance in analyzing high-dimensional data such as images and video. This scalability, combined with increasing computational power, positions DL as a transformative tool for fall detection and other complex tasks. Additionally, techniques such as hyperparameter tuning—adjusting factors, viz. learning rate, batch size, and architecture—further optimize performance, enabling these models to achieve state-of-the-art results in high-stakes applications.

1.6.1 Supervised Learning

Supervised learning is a machine learning paradigm in which models are trained on labeled datasets, where each sample is paired with a known output. This approach is commonly used in fall detection systems to classify actions as falls or non-falls based on labeled examples. The model learns patterns by minimizing the error between its predictions and the provided labels, enabling it to generalize to new, unseen data [61]. A significant challenge in supervised learning for fall detection is the need for large, diverse, and high-quality labeled datasets. Annotating such datasets is labor-intensive and costly, particularly for thermal data, where expert knowledge is often required [57]. Despite these challenges, supervised models excel in scenarios where comprehensive datasets are available, achieving high accuracy by associating new inputs with patterns learned during training.

1.6.2 Unsupervised Learning

Unsupervised learning involves training models on unlabeled datasets, allowing them to discover patterns, structures, or relationships within the data without the need for explicit labels. In fall detection, this approach is particularly advantageous for identifying anomalous or unexpected behaviors, such as falls, in situations where labeled datasets are limited or costly to obtain. By leveraging this method, unsupervised models can effectively learn to differentiate between normal activities and potential fall events, even in the absence of extensive labeled data. Common tasks in unsupervised learning include clustering, which groups data based on similarities, and anomaly detection, which identifies deviations from typical patterns. These methods can complement supervised learning by leveraging large amounts of unlabeled data to uncover insights that guide further analysis. However, the absence of labeled data poses challenges in evaluating unsupervised models, as there is no explicit ground truth to measure performance.

1.6.3 Types of Models

A variety of DL models can be used to address domain-specific challenges. The following section highlights the rationale behind the selection of 3D Convolutional Neural Networks (3D CNNs) and Autoencoders (AEs) as the primary architectures for this thesis, based on an extensive review of the literature and the unique demands of fall detection systems. A visual representation of each of these models can be found in Figure 1.6, with further explanations for these models provided in Chapter 3.

Model Selection Process

The selection of 3D CNNs and AEs was guided by an evaluation of models commonly used in fall detection and related fields, such as video anomaly detection and action recognition. Among these, 3D CNNs, AEs, and Transformers emerged as the leading candidates due to their ability to process spatiotemporal data. However, Transformers, while highly effective at capturing long-term temporal dependencies, were excluded as they are less suitable for short-duration events,

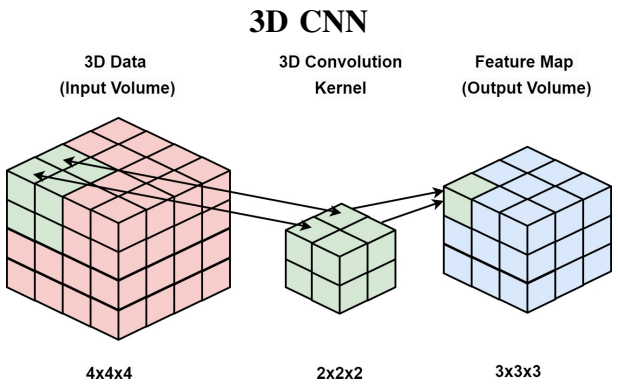


Illustration of a 3D Convolutional Neural Network model, showing the convolution operation from input data to feature map generation. (cf. Section 3 for more information on 3D CNNs).

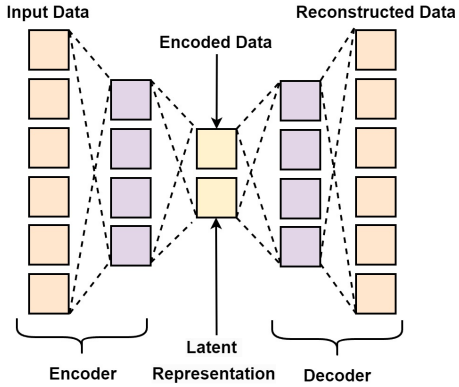


Illustration of an Autoencoder model, showing the encoder, latent representation, and decoder processes for data reconstruction. (cf. Section 3 for more information on AEs).

Figure 1.6: Visual representations of the core models used in this thesis.

such as falls, which typically span only a few frames. This limitation rendered Transformers impractical for the scope of this work. Further exploration of models used in general video analysis revealed no alternative architectures that matched the requirements of thermal fall detection. The spatiotemporal complexity of fall detection—requiring the simultaneous analysis of spatial and temporal features—posed additional constraints, as many traditional models cannot effectively process such data. As a result, 3D CNNs and AEs were identified as the most suitable architectures for addressing the challenges of thermal fall detection.

3D Convolutional Neural Networks

3D Convolutional Neural Networks (3D CNNs) are a specialized type of supervised machine learning model designed to analyze volumetric data or sequences of images [62]. Unlike traditional 2D CNNs, which capture only spatial information within a single image, 3D CNNs simultaneously process spatial and temporal information, making them ideal for video analysis and tasks where time-based context is critical. The core operation in a 3D CNN is the 3D convolution, where a filter slides across the width, height, and depth (or time) of the input data, generating feature maps that

preserve spatial and temporal details. Pooling layers further reduce the dimensionality of these feature maps, retaining the most significant information while improving computational efficiency. In a typical 3D CNN architecture, early layers capture basic motion features like edges and textures, while deeper layers learn more complex patterns, such as the progression of a fall [61, 63].

3D CNNs are particularly well-suited for thermal fall detection, where temporal sensitivity is crucial. Falls occur over short periods, and the network must distinguish subtle changes in motion across consecutive frames from normal activities like sitting or walking. By analyzing sequences of thermal images, 3D CNNs effectively capture the temporal dynamics of a fall, achieving greater accuracy compared to 2D CNNs, which process each frame independently and lack temporal context [62, 64]. Additionally, thermal-based 3D CNN systems offer inherent privacy advantages. Unlike RGB systems that capture identifiable visual data, thermal imaging obscures personal details while still providing rich information for detecting motion patterns. This makes thermal 3D CNNs especially suitable for privacy-sensitive environments like senior living facilities, where preserving dignity is paramount [57]. Broadly, CNNs—including 3D CNNs—are widely regarded as the gold standard for vision-based systems, consistently outperforming traditional computer vision methods in tasks like object classification, detection, and segmentation [60]. By extending these capabilities to video data, 3D CNNs provide a robust solution for fall detection and other time-sensitive applications.

Autoencoders

Autoencoders (AEs) are unsupervised neural networks designed for tasks such as data compression, noise reduction, and anomaly detection. They consist of an encoder, which compresses input data into a lower-dimensional latent space, and a decoder, which reconstructs the input from this representation. The model learns to minimize the difference between the original input and the reconstructed output. In thermal fall detection, AEs are effective for anomaly detection by training on normal movements (non-fall events) and learning to reconstruct these patterns with high accuracy. During inference, falls or other anomalies that deviate significantly from the learned patterns result in high reconstruction errors, signaling abnormal events. This approach is particularly useful

in scenarios where labeled fall data is scarce, offering a privacy-preserving and efficient solution for detecting falls in thermal imaging data.

In summary, the selection of 3D CNNs and Autoencoders as the primary architectures for this thesis was driven by their ability to address the spatiotemporal complexities inherent in fall detection. 3D CNNs excel in supervised settings, leveraging their capacity to capture both spatial and temporal features across brief sequences, while AEs provide a robust unsupervised approach for detecting anomalies in scenarios with limited labeled data. Together, these models form a comprehensive foundation for developing accurate and privacy-preserving thermal fall detection systems.

1.6.4 Evaluation Metrics

Evaluation metrics play a critical role in fall detection research, ensuring that model performance is rigorously assessed and comparable across studies. However, no single "ideal" metric exists, as different approaches prioritize varying outcomes. For example, unsupervised models often rely on reconstruction error thresholds to identify falls, balancing sensitivity (true positive rate) and specificity (false positive rate). Metrics such as the Receiver Operating Characteristic Area Under the Curve (ROC-AUC) are commonly used in this context to evaluate the trade-off between these two factors [11]. Relying exclusively on metrics like ROC-AUC, however, can be misleading. While a high ROC-AUC score suggests strong overall performance, it does not necessarily reflect balanced outcomes between the positive and negative classes (i.e., falls and non-falls). Models with high sensitivity but frequent false positives may appear effective in controlled evaluations but perform poorly in real-world deployments, where excessive false alarms can waste resources and erode user trust [8, 18]. Similarly, prioritizing fall detection rates without addressing false alarms can result in systems that are impractical for deployment.

Recognizing these limitations, this work adopts a holistic evaluation approach by reporting a comprehensive set of metrics tailored to the binary classification nature of fall detection. Choosing a wide arrangement of metrics ensures a balanced assessment of a model's ability to detect falls while minimizing false alarms. This is particularly critical in fall detection, where both false nega-

tives (missed falls) and false positives (false alarms) can have severe consequences for user safety and system reliability. By emphasizing diverse evaluation metrics, this thesis aims to facilitate fair and meaningful comparisons between models, improve the robustness of fall detection systems, and advance the development of solutions suitable for real-world deployment. This work utilizes the ROC-AUC metric defined in (1.1) as an evaluation metric for the proposed model.

$$\textit{ROC-AUC} = \int_{-\infty}^{\infty} ROC(\tau)\partial\tau \tag{1.1}$$

Here, $ROC(\tau)$ represents the ROC curve at a given threshold value τ, and $d\tau$ denotes the differential element in the integral. The ROC curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR), defined as follows:

$$\text{True Positive Rate (TPR)} = \frac{TP}{TP + FN} \tag{1.2}$$

$$\text{False Positive Rate (FPR)} = \frac{FP}{FP + TN} \tag{1.3}$$

In (1.2) and (1.3), TP (True Positives) and FP (False Positives) represent samples that are correctly and incorrectly classified as positive, respectively. Similarly, TN (True Negatives) and FN (False Negatives) represent samples that are correctly and incorrectly classified as negative, respectively.

Accuracy, defined in (1.4), measures the overall correctness of the model's predictions.

$$\textit{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1.4}$$

The F1-Score, defined in (1.5), is the harmonic mean of precision and recall, providing a balanced measure that accounts for both false positives and false negatives.

$$\textit{F1-Score} = 2 \times \frac{\textit{Precision} \times \textit{Recall}}{\textit{Precision} + \textit{Recall}} \tag{1.5}$$

Finally, the Matthews Correlation Coefficient (MCC), as defined in (1.6), provides a comprehensive metric that takes into account all four confusion matrix components, offering a more balanced evaluation, particularly in cases of class imbalance.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{1.6}$$

By incorporating these metrics, this work ensures a robust and comprehensive evaluation of a given model's performance, capturing various aspects of binary classification that are crucial for accurate and reliable predictions.

1.7 Thesis Contribution

This thesis presents a comprehensive framework for thermal fall detection, bridging the gap between controlled laboratory research and real-world deployment. By addressing critical limitations in existing datasets, architectures, and methodologies, this work reorients the field toward practical, deployable solutions designed to meet the needs of seniors and industry stakeholders. The key contributions of this thesis are as follows:

- **Insights from a Hybrid Learning Approach:** This thesis investigates a hybrid architecture that combines supervised and unsupervised learning paradigms via a stacking meta-model. While the hybrid approach does not feature in the final fall detection system, its exploration provided invaluable insights into addressing small, less representative datasets. These findings informed the development of the TF-66 dataset and the brute-force data generation approach, both of which proved critical in enabling robust supervised learning for fall detection. The hybrid model also serves as a foundational strategy for future researchers working with constrained datasets (cf. Chapter 3).

- **Creation of the TF-66 Dataset:** A cornerstone of this thesis is the creation of TF-66, a robust, diverse, and privacy-preserving dataset for thermal fall detection. TF-66 surpasses existing datasets in its participant diversity, environmental variety, and real-world applica-

bility. This dataset establishes a benchmark for meaningful comparisons of fall detection models while emphasizing deployability and usability (cf. Chapter 4).

- **Advancing Data Generation Techniques:** This thesis introduces innovative data generation methodologies that significantly enhance model training and evaluation. The data generator provided in Chapter 4 allows prospective researchers to easily develop balanced batches of data for training, mitigating class imbalance issues. Researchers can train and evaluate models on specific subsets of the dataset by simply toggling a single parameter, enabling the creation of personalized models tailored to diverse scenarios. Additionally, the sequence length of generated samples can be customized with ease, further enhancing the flexibility of the framework. These methodologies offer a reproducible, scalable approach to generating representative and customizable samples, fostering advancements in fall detection research.

- **Development of State-of-the-Art Models:** The thesis refines the supervised learning approach, culminating in the development of models that achieve state-of-the-art performance on the TF-66 and TSF datasets. The ConvLSTM + Optical Flow model represents the optimal architecture for TF-66, while the BiConvLSTM + Attention Mechanisms model sets a new benchmark on TSF, achieving an unprecedented ROC-AUC of 99.7%. These models highlight the importance of leveraging spatiotemporal and attention mechanisms in fall detection (cf. Chapter 5).

- **Establishing a New Research Benchmark:** By combining TF-66 with comprehensive evaluation frameworks, this thesis addresses the lack of standardization and generalizability in fall detection research. TF-66 provides a robust foundation for future studies, enabling fair comparisons of models and catalyzing progress in the field.

- **Real-World Focus and User-Centered Design:** Incorporating feedback from seniors, the primary end-users, ensures that the developed solutions prioritize usability, accuracy, and privacy. This user-centered design focus bridges the gap between academic research and market-ready systems, emphasizing the importance of practical deployment.

These contributions push the boundaries of fall detection research by delivering robust datasets, flexible data generation techniques, and cutting-edge models. This thesis offers a practical guide for future researchers to develop accurate, scalable, and privacy-preserving fall detection systems that are ready to make a real-world impact.

Chapter 2

Literature Review

2.1 Overview

This chapter provides a comprehensive review of existing research and literature in the field of fall detection, identifying critical gaps and outlining how this work addresses these limitations. The discussion begins by defining the characteristics of an "ideal" fall detection device, informed by insights from existing literature and user preferences. Next, the chapter examines existing thermal fall detection datasets, highlighting their strengths and limitations. A historical perspective on fall detection research is then presented, followed by an in-depth review of various approaches, including traditional machine learning methods, wearable solutions, ambient systems, vision-based techniques, and ensemble models. By synthesizing these findings, this chapter establishes the foundation for the novel contributions of this study.

2.2 Features of an Ideal Fall Detection Device

This section explores the characteristics of an ideal fall detection system to guide research efforts toward the development of effective and practical solutions. The absence of a standardized definition for an optimal fall detection system (FDS) in the literature has led to varied methodologies and inconsistent evaluations, creating challenges in comparing and advancing proposed systems.

To address this gap, this section synthesizes user requirements and insights from previous studies, identifying the key features that an ideal FDS should possess to ensure usability, accuracy, and real-world applicability.

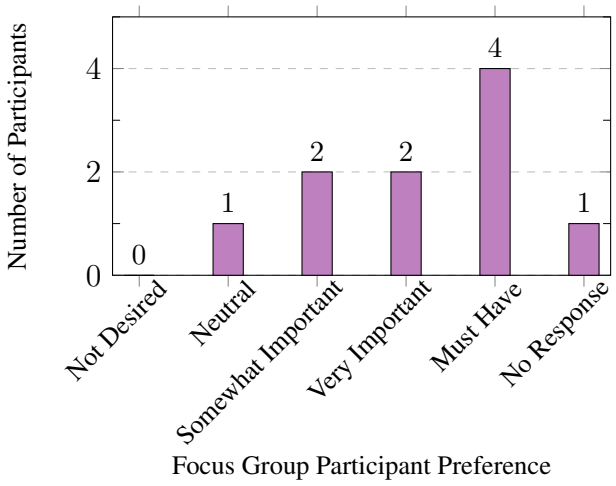
2.2.1 A Study of User Preferences: from Literature and Focus Groups

Prioritizing user preferences is essential to ensuring both the adoption and effectiveness of fall-detection devices. While focus group studies have examined seniors' perspectives on these technologies, the results have been inconclusive, often reflecting biases in the presentation of the technologies [18, 65–67]. For instance, a 2009 study found that 96% of participants were receptive to FDS [65], while a 2013 study revealed skepticism among potential users [67]. This variation may stem from differences in how participants are introduced to the technology and the moderators' framing of questions. Despite these inconsistencies, certain trends can be extracted. Independence is a key concern for seniors, many of whom express a preference for non-wearable, discreet devices that do not stigmatize aging or signal a loss of independence [13, 35]. Privacy is also a significant factor, as some users are uncomfortable with constant monitoring or intrusive solutions [13, 15]. The previously mentioned 2015 study [18] reported that while 96% of participants were open to video monitoring, only 48% would use such systems due to privacy concerns [66]. Seniors also emphasize the importance of reducing false alarms, which can lead to device abandonment if not addressed [13, 18, 39, 53]. In terms of system usability, seniors prefer automatic fall detection solutions that do not require manual activation. Systems that require users to press a help button after a fall are often deemed unreliable, as participants may forget to do so, especially in critical situations [18, 26, 68]. Affordability is another critical consideration, with seniors expressing a preference for systems that are both effective and reasonably priced [13, 18]. These mixed findings suggest that although user preferences can vary, certain common themes like privacy, ease of use, and reliability must be addressed. However, the inconsistencies and biases present in past focus groups—such as how new technologies are introduced and understood by seniors—indicate the need for more controlled and unbiased studies.

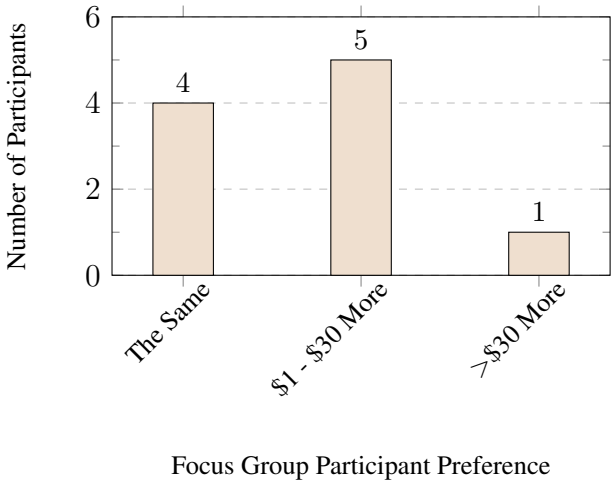
2.2.2 New Focus Group Insights

Recognizing the limitations of past focus group studies, a new focus group was conducted at a local retirement facility to gather unbiased insights into the preferences of fall detection devices. Ten elderly female residents participated, and the session was led by neutral moderators to avoid influencing responses. The participants were encouraged to provide open feedback on a range of fall detection system features, including privacy, ease of use, and cost. This focus group aimed to minimize bias by ensuring participants fully understood the key functionalities of modern FDS and by collecting feedback anonymously. Structuring the session in this way provided more reliable and representative data regarding the features seniors truly value. The insights gathered will help define the key characteristics needed for an ideal fall detection device. One of the primary findings was the importance of privacy. As shown in Figure 2.1a, eight out of nine participants indicated that privacy-preserving features are at least somewhat important, with six rating privacy as "very important" and four considering it a "must-have" feature for any fall detection system. Similarly, affordability also emerged as a critical issue. As seen in Figure 2.1b, eight out of nine participants indicated that they would be willing to pay up to $30 more per month for a thermal, hands-free fall detection system, compared to existing pendant-based solutions. Only one participant expressed a willingness to pay significantly more than $30 per month.

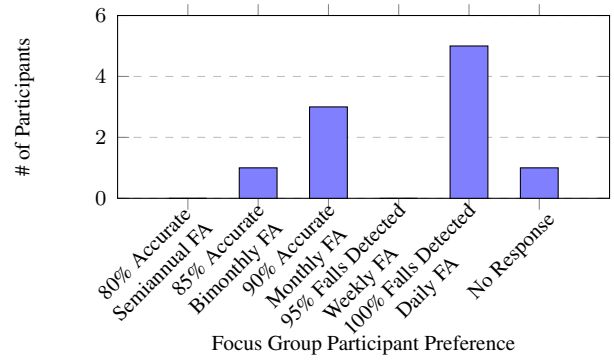
During the focus focus group, one of the distributed handouts, depicted in Figure 2.1d, aimed to gauge potential user behavior and their tolerance for false alarms in balancing detection accuracy and user annoyance. The summarized results, illustrated in Figure 2.1c, show a nearly equal division between participants who prioritize high accuracy despite the risk of false alarms, and those who prefer fewer false alarms at the cost of reduced accuracy. A subsequent question explored whether alerts should be triggered only if someone remains prone for a prolonged period or also if they quickly recover and leave the scene. The responses mirrored the initial findings, indicating varied preferences among potential users. Market acceptance of future FDS requires addressing diverse individual preferences, as a one-size-fits-all approach risks neglecting significant user needs. The focus group identified privacy, affordability, and ease of use as key priorities for seniors. While these insights offer valuable guidance on what end users prioritize, it is equally



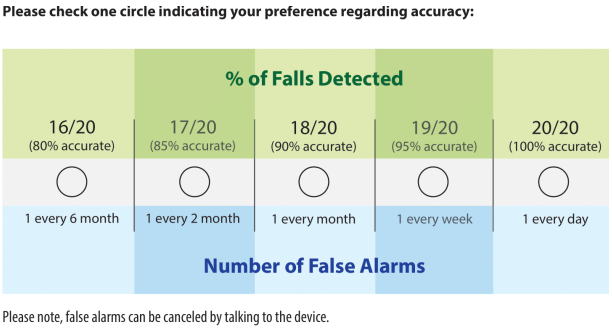
(a) Preferences of an FDS protecting user privacy.



(b) Preferences for the price of a thermal FDS compared to existing pendant-based solutions.



(c) Accuracy/false alarm (FA) trade-off preferences.



(d) Handout given to the focus group's participants.

Figure 2.1: Results of the focus group research in TF-66.

important to consider the broader trends in the fall detection literature. In the next section, the key recommendations from the academic community will be explored to ensure that the development of an ideal fall detection system balances both user preferences and expert advice.

2.2.3 Trends in the Literature

The literature identifies several critical factors for successful FDS: enhancing user compliance through passive devices, ensuring privacy, minimizing false positives, and reducing device visibility [11, 15, 18, 39, 58, 69–71]. However, significant challenges persist, including small sample

sizes and the lack of standardized benchmark datasets, which impede objective comparison across studies [18, 33, 72, 73]. Another prevalent issue is the reliance on private, custom-made datasets, limiting the ability to compare models trained on varying data that cannot be obtained [71]. Many solutions are also optimized for controlled lab environments to achieve impressive performance metrics, often at the expense of real-world applicability. For example, these systems may rely on a large number of concurrent sensors, expensive high-resolution devices, or overly complex models that cannot operate in real time. Addressing these challenges and aligning system designs with real-world constraints should be the priority. The following sections will examine the specific features required for an ideal fall detection system, considering these overarching trends.

2.2.4 Optimal Sensor Modality

Selecting the optimal sensor modality for fall detection is critical, with the literature generally favoring privacy-preserving, vision-based sensors. It seems the best type of sensors are vision based sensors rather than wearable or ambient solutions, as long as privacy considerations can be integrated into the system [74]. Rather than relying on facial information, effective fall detection primarily depends on spatial and temporal motion of the human body [75]. Thermal and infrared sensors offer significant advantages over RGB sensors by capturing heat signatures, thereby preserving privacy, functioning effectively in low-light environments, and minimizing irrelevant background clutter [7, 15, 20, 39, 70].

Advantages of Thermal Sensors

Thermal sensors offer robust generalization in machine learning model inference as they are less affected by the environment [62], unlike RGB-based solutions which are affected by lighting and background variations. These factors necessitate environment-specific training data for RGB systems, as differences in room layout between training and deployment can lead to excessive false alarms [39]. Thermal sensors, however, capture only heat-emitting objects, ensuring consistency across different settings and reducing the need for personalized data. This consistency makes ther-

mal sensors ideal for widespread deployment, especially in residential environments with varying lighting and furniture layouts.

Limitations of Thermal Sensors

Thermal sensors are not without limitations. Background artifacts, such as variations in room temperature, can introduce clutter into the image and reduce detection accuracy [70, 76]. Additionally, smaller heat-emitting objects, such as pets or cups filled with hot liquid, may trigger false alarms as they are captured by thermal sensors. These challenges can be addressed by integrating person detection directly into a fall detection system to differentiate between human and non-human heat sources [19]. While object detection has been employed to localize humans, such techniques add unnecessary complexity and may affect real-time processing on embedded systems [70].

Alternative Modalities

Depth sensors, such as the Kinect, have also been explored due to their privacy-preserving nature. However, they often suffer from resolution degradation at longer distances, struggle in sunny environments, and produce data with "holes" caused by sensor limitations [22, 30, 33, 39, 55, 77]. Moreover, these sensors are not widely accepted in real-world settings, such as long-term care homes, due to health concerns [22]. The infrared portion of Kinect images appears illuminated by visible light, compromising privacy and making them unsuitable for real-world applications. RGB-based systems that extract optical flow information to track motion between frames have also been studied [10, 30]. While this method addresses some privacy concerns by ignoring static background data, it remains highly sensitive to lighting changes, making it less reliable in dynamic environments [30]. If using optical flow as an input modality in a model, thermal sensors are preferable as the input data due to their robustness against lighting fluctuations. However, while thermal imaging optical flow solutions offer advantages over RGB optical flow solutions, the use of optical flow could increase computational demands, which may limit its feasibility on embedded devices [7, 30].

Light Detection and Ranging (LiDAR) has also been explored, but it is seen as too expensive to be an effective real-world solution for fall detection [78]. Ultra-Wideband radar and WiFi-based systems have recently gained traction in the field of fall detection due to their ability to mitigate adherence issues associated with wearable solutions while preserving privacy. However, these technologies remain suboptimal for the critical demands of fall detection. UWB systems are prone to environmental artifacts, such as reflections from walls and furniture, which can result in false positives and necessitate extensive calibration for each deployment [9]. Similarly, WiFi-based systems are highly susceptible to interference from other devices and environmental changes, significantly undermining their reliability in real-world settings [50]. Furthermore, both technologies struggle to identify specific individuals in multi-occupant environments, such as long-term care facilities, further limiting their practicality for widespread use.

The widespread availability and low cost of RGB cameras have led to efforts to mask facial regions in RGB data to preserve privacy while using the images for fall detection models. However, the effectiveness of facial masking diminishes with increasing distance from the sensor, and model inversion attacks can reconstruct private features from the dataset, ultimately compromising privacy [79, 80]. Even if one could somehow guarantee privacy, the use of RGB cameras introduces a critical limitation that cannot be overlooked. The methods for fall detection require large datasets, but because each environment appears unique to an RGB camera, exponentially more data is needed to ensure accurate detection in unseen settings—an essential requirement for life-saving deployments. This limitation means that each recording environment would necessitate its own personalized training dataset, a requirement that is neither practical nor scalable for real-world installation.

Given these concerns, thermal sensors, which inherently obscure identity, remain the most reliable and secure modality for fall detection. Their robustness against lighting fluctuations, ability to operate in low-visibility conditions, and their intrinsic privacy-preserving nature make them a superior choice over alternative modalities.

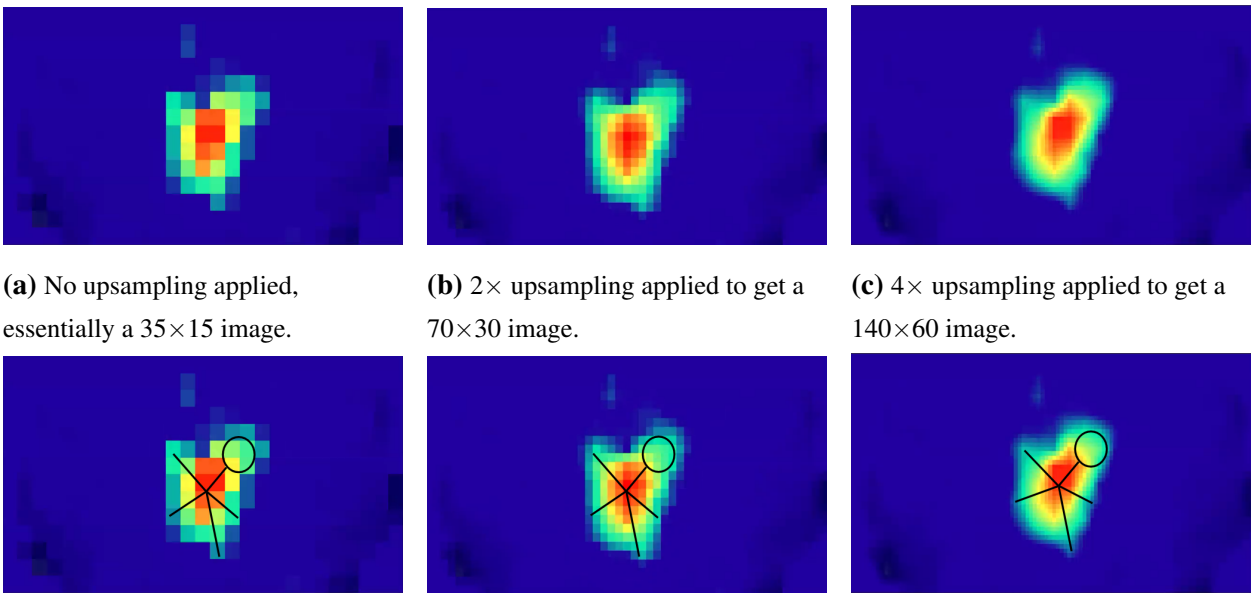
Single-Modality vs. Multi-Modality Approaches

While combining multiple sensor modalities might seem advantageous from a technical perspective, it significantly increases both the cost and complexity of the system [39, 81]. Synchronization issues between different modalities further complicate real-time deployment, making a single-modality solution—such as thermal imaging—more practical for cost-effective and reliable fall detection [22, 81]. While some seniors are concerned about thermal sensors due to the misconception that they are being watched, studies show that small levels of intrusiveness are acceptable if the system proves its value [13, 82, 83]. Thus, with appropriate user education, thermal fall detection offers an ideal solution for real-world deployment, balancing privacy, accuracy, and practicality.

2.2.5 Need for a High Resolution Sensor

For vision-based FDS to maximize fall detection rates, the use of high-resolution sensors is essential. Higher resolution provides more discriminative features for machine learning models to analyze, enhancing their learning and detection capabilities. Conversely, sensors with lower resolution tend to merge heat signatures into indistinguishable "blobs". This aggregation makes it challenging to discern specific postures, such as distinguishing between a fall, a bend, or a reclined position. Given the complex and rapid nature of falls, high-resolution sensors are crucial for accurate and effective fall detection. When using low resolution sensors, it has been proven to be difficult to detect small motions because of the lack of difference in the resulting low-resolution images [84]. This is demonstrated through a visual example found in Figure 2.2 which displays images of an individual laying on their stomach on the ground with different sensor resolution options in each column of subfigures. Each image is recorded using the Calumino Thermal Sensor Evaluation Kit v3.1 sensor, with different in-application upscaling options. Figures 2.2a, 2.2b, and 2.2c display images with a $1\times$, $2\times$, and $4\times$ upscaled value, respectively, meaning the image resolution for each subfigure is 35×15, 70×30, and 140×60, respectively.

In Figure 2.2f, a clear outline of a person lying on the ground beneath the sensor is visible. When comparing this to Figure 2.2c, it becomes apparent where the hand and feet outlines are



(a) No upsampling applied, essentially a 35×15 image.

(b) 2× upsampling applied to get a 70×30 image.

(c) 4× upsampling applied to get a 140×60 image.

(d) No upsampling applied, a 35×15 image, stick figure annotated.

(e) 2× upsampling applied to get a 70×30 image, stick figure annotated.

(f) 4× upsampling applied to get a 140×60 img, stick figure annotated.

Figure 2.2: Six images of a participant laying on their stomach.
Note: The first row displays the raw sensor feed, and the second row adds stick figure annotations to highlight the participant's position. Columns display 1×, 2×, and 4× upscaling.

located in the 4× upscaled image. The difference between these limb outlines in the 4× and 2× upscaled images highlights how a lower upscaling setting diminishes the clarity of certain features. For example, in Figure 2.2b, the participant's left leg is difficult to distinguish. Similarly, reducing the upscaling to 1× in Figure 2.2a makes it hard to detect the right arm or left leg, and the head is no longer displayed as a smooth, rounded shape. This demonstrates the importance of using a sensor with a higher resolution to maintain critical image details.

2.2.6 Sensor Placement and Occlusion Mitigation

Sensor placement is one of the primary challenges in vision-based FDS. Wall-mounted RGB cameras frequently encounter occlusion issues, where objects block the sensor's view, resulting in undetected falls [7, 55, 73, 81]. While minor occlusions in datasets can enhance generalizability, significant occlusions—such as falls occurring behind large objects like couches—pose critical

risks in real-world scenarios, potentially leading to life-threatening failures. Therefore, minimizing occlusions is essential to ensure the reliability and effectiveness of FDS [7, 81].

Additionally, variations in sensor height and angle can impact detection accuracy, particularly when individuals fall behind furniture or other obstructions [8, 81]. Wall-mounted sensors also struggle with depth perception, making it challenging to detect falls accurately when individuals land on furniture [85]. Performance further deteriorates when sensors are mounted in room corners due to distorted images caused by unusual angles [78] and the presence of "blind spots" directly beneath the sensor [52]. Addressing these placement challenges is critical for designing robust and practical FDS. Thus, the ideal solution is ceiling-mounted sensors, which provide a consistent overhead view, minimizing occlusion issues and enhancing generalizability across environments [55, 86, 87]. This setup improves detection accuracy regardless of room layout, unlike wall-mounted sensors. Ceiling-mounted sensors also offer precise distance estimation with median errors as low as 0.07 meters [19, 88], and incorporating room height and distance data into models can help distinguish falls from non-fall actions, reducing false alarms. Additionally, ceiling-mounted sensors with wide fields of view can cover larger areas, reducing the number of devices needed and lowering system costs.

2.2.7 Accessibility and Price of Device

Affordability and accessibility are crucial factors in the adoption of FDS. Existing thermal FDS often use high-resolution or multimodal sensors to enhance detection accuracy [39]. While these features improve performance, they significantly increase the cost of the system, limiting the technology's accessibility to a smaller, more affluent segment of the population. The literature review highlights the trade-off between sensor resolution and cost. For a fall detection device to become a gold standard, it must be affordable to the average consumer. Currently, pendant-based systems are priced around $40 to $50 per month depending on features [19]. While concerns have been raised about the higher costs of thermal sensors, previous research has demonstrated that a thermal-based fall detection system can be developed for as little as $150. However, this solution utilized a combination of motion, floor, and low-resolution thermal sensors, relying on multiple modalities to

function effectively [19]. This highlights the feasibility of an affordable thermal-based solution, making it more accessible for widespread deployment in real-world settings without compromising detection accuracy.

2.2.8 Personalization and Environmental Diversity in Fall Detection Devices

A 2015 study [89] using National Electronic Injury Surveillance System data found that the location of falls varied significantly by age and sex. The majority of falls occurred in the bedroom, with this proportion increasing from 19.8% for individuals aged 65-74 to 31.6% for those aged 85 and older. Other common fall locations included stairs, bathrooms, kitchens/dining rooms, and living rooms. Women were more likely to fall in the bedroom (2.2% higher) and kitchen (2.5% higher), while men had a higher incidence of falls on stairs (0.5% higher) [89].

These findings highlight the need for personalized FDS tailored to an individual's environment. Falls in bedrooms differ significantly from those in other areas due to variations in furniture and layout, highlighting the need for a robust fall detection system that incorporates data from a diverse range of home environments [33]. Ceiling-mounted thermal sensors, which can accurately estimate distances to individuals, require training on data from environments closely resembling the intended installation settings. This personalized approach enhances classification accuracy and overall system performance.

A major limitation of existing datasets is lack of recordings from multiple environments, which reduces their generalizability to real-world scenarios. Variations in sensor placement, particularly ceiling or wall-mounted heights, can distort the perceived size of individuals in recordings. Sensors mounted higher make individuals appear smaller, while closer placements enlarge their perceived size. To minimize false alarms and improve accuracy, datasets must capture diverse environments with varying room heights, furniture arrangements, and layouts. While some researchers suggest using CNNs to address environmental variations, relying solely on CNNs may not be sufficient. A more effective solution involves creating a training dataset that incorporates data from multiple environments, ensuring robustness without over-dependence on CNNs to compensate for dataset deficiencies. Ignoring room height variations during dataset creation can lead to misinterpreta-

tions during real-world inference procedures on new, unseen data, particularly when systems are deployed in spaces with varying dimensions. For example, a system may incorrectly classify a small individual as someone lying on a bed due to perceived size differences influenced by sensor placement.

Thus, drawing insights from focus groups and literature, an ideal fall detection device should have the following key features: (i) Be non-wearable and passive, requiring no interaction, (ii) Preserve privacy using thermal sensors or other non-intrusive modalities, (iii) Detect falls reliably while minimizing false alarms, (iv) Use high-resolution sensors adaptable to diverse environments and room configurations, and (v) Ensure cost-effectiveness for accessibility and adoption.

2.3 Datasets

2.3.1 Existing Datasets

A significant portion of the fall detection research relies on datasets that are not publicly available. This section focuses on publicly available datasets, which provide researchers with opportunities to compare models directly, enabling a fair evaluation of different solutions to identify the most optimal approaches. As discussed, an ideal fall detection system should utilize ceiling-mounted thermal sensors, operate effectively across diverse environments, include varied participants, and enable real-time detection. However, existing datasets have several recurring limitations:

1. **Limited Environment Diversity**: Most datasets are recorded in controlled settings with uniform conditions, which restricts their generalizability to real-world scenarios.

2. **Inadequate Participant Representation**: Falls are often simulated by younger participants, failing to capture the slower and more unsteady dynamics of elderly falls, leading to reduced performance in practical applications.

3. **Occlusion and Sensor Placement Issues**: Wall-mounted sensors encounter occlusion challenges, while ceiling-mounted sensors, which mitigate such issues, are seldom used.

4. **Privacy Concerns**: Many datasets rely on RGB sensors, compromising user privacy. Although thermal sensors preserve privacy, they remain underutilized.

5. **Unrealistic Fall Scenarios**: Simulated falls by younger individuals often lack the hesitation or imbalance characteristic of falls by elderly individuals.

These limitations highlight the pressing need for a dataset that addresses these gaps by incorporating diverse environments, realistic fall scenarios with elderly participants, privacy-preserving thermal sensors, and reliable ceiling-mounted setups. Table 2.1 provides a comparative analysis of publicly available thermal fall detection datasets against the ideal criteria for an FDS. The datasets are evaluated on factors such as resolution, participant diversity, recording environments, sensor placement, privacy considerations, frame rates, and the use of multimodal sensors. Each dataset listed in Table 2.1 is further detailed in the subsequent sections, outlining its key features, strengths, and limitations in the context of fall detection research.

Table 2.1: Comparative analysis of various publicly available thermal fall detection datasets with a minimum resolution of 16×16
Note: Sample Size - number of participants, R - recording environments, C - ceiling-mounted sensors, P - privacy-preserving images, ADL - Activities of Daily Living recorded, Indoor - recorded indoors, FPS - frames per second, SM - reliance on thermal sensors alone or inclusion of other modalities

Dataset	Resolution	Sample Size	#Falls	#R	C	P	ADL	Indoor	FPS	SM
TSF [70] (2018)	640×480	1	35	1	×	×	✓	✓	12	✓
eHomeSnr [20] (2019)	32×24, 1×8	6	448	1	×	✓	×	✓	16,5	×
UP-Fall [55] (2019)	NA	17	222	1	×	×	✓	✓	18	×
MUVIM [39] (2022)	1440×1080	30	400	1	✓	×	✓	✓	8.7	×
TMF [90] (2022)	384×288	0	22	1	×	×	×	×	25	✓

Evaluating Existing Datasets Against Ideal Criteria

eHomeSeniors Dataset [20]: It contains 448 falls captured from six participants using a wall-mounted sensor configuration. However, the resolution is extremely low (32×24), and additional sensors with even lower resolutions (1×8) were used to detect the transition from standing to ground level. Occlusion issues due to wall-mounted sensors and a single-room recording further limit its usability in broader contexts.

TMF Dataset [90]: This dataset stands out for its innovative exploration of outdoor fall detection using thermal sensors. However, it is significantly constrained by the fact that only 22 falls were recorded, all simulated using a heated mannequin instead of real human subjects. While the use of a mannequin is creative, the lack of genuine human falls limits the dataset's relevance. Additionally, its outdoor focus reduces its applicability to indoor environments, which represent the majority of real-world use cases for FDS.

The following paragraphs discuss the three most commonly used datasets in thermal imaging-based fall detection–TSF, UP-Fall, and MUVIM in greater detail.

TSF Dataset [70]: It employs a high-resolution FLIR camera (640×480), but it is restricted to a single recording environment and includes a relatively small sample size of 35 falls. The inclusion of visible light in the dataset compromises participant privacy, a critical concern for fall detection in sensitive settings. Additionally, the use of a wall-mounted sensor introduces occlusion issues, which could hinder the system's effectiveness in real-world applications where objects and furniture may block the sensor's view.

UP-Fall Dataset [55]: It employs a multimodal approach, incorporating thermal sensors, RGB cameras, wearable sensors, and an EEG headset. While this diversity allows for comparisons among sensor types, it complicates the focus on optimizing fall detection modalities. The unspecified-resolution thermal sensors primarily serve for interruption detection rather than direct fall recognition. Six infrared sensors, positioned 0.4 meters above the floor, concentrate on detecting human presence rather than actual falls. Recorded in a controlled environment with constant lighting, the dataset's applicability to real-world scenarios is limited, as lighting conditions can vary widely [55]. In testing, infrared sensors detected fewer than one in three falls, and synchronization issues with other sensors required upsampling or frame repetition. While UP-Fall supports testing multiple sensors, its multimodal design and focus on presence detection limit its effectiveness for developing a cost-efficient, real-time fall detection system. The lack of ceiling-mounted sensors further restricts its ability to reduce occlusion and improve fall recognition.

MUVIM Dataset [39]: It consists of 400 falls recorded from 30 participants using high-resolution (1440×1080) thermal cameras. While promising for detailed analysis, privacy concerns arise from thermal-based sensors that do not fully obscure identity, due to the incorporation of visible light to the thermal images. The dataset's reliance on six concurrent sensors and a wearable device makes it costly and impractical for mass adoption in residential environments. Further limitations include the use of depth cameras, which are often unreliable for fall detection [22, 30, 39, 55]. None of the participants were over 30, limiting the dataset's applicability since elderly individuals—who fall more slowly—are the primary demographic for FDS [20]. Additionally, the forward looking infrared (FLIR) cameras used lacked detail and frequently failed to capture entire fall events. Freezing during trials resulted in data loss, leaving only 182 falls for testing and 62 videos of activities of daily living (ADLs) for training [39]. This small dataset hinders accurate model comparisons and generalization, as some common activities may be underrepresented. Although MUVIM achieved a promising ROC-AUC score of 0.88 [39], the dataset's dynamic rescaling of thermal images complicates analysis. When a person enters or exits the frame, color intensity adjusts, causing the same room to appear drastically different across varying configurations. Without a standardized color range, fall detection models struggle to generalize, resulting in inaccurate predictions in real-world implementations.

Besides employing the aforementioned thermal fall detection datasets, recent fall detection research has treated the fall detection problem as anomaly detection, employing unsupervised learning methods like AEs trained on ADL samples [39, 75, 76]. In this framework, any detected anomaly is classified as a fall. While promising, this approach necessitates a comprehensive thermal dataset that captures diverse actions to refine AE models. The Thermal-IM dataset [91], offers 783 video clips and approximately 560,000 non-fall frames, but it lacks essential fall samples required for system performance testing, leaving a gap for specialized thermal fall detection datasets. Recent research has shown that combining supervised and unsupervised learning significantly improves fall detection performance [92]. This underscores the limitations of relying solely on anomaly detection, given the precision needed to maximize true fall detections while minimizing false positives. The complementary use of AEs trained on ADL samples alongside supervised

models trained on labeled fall data emphasizes the need for a benchmark dataset. Such a dataset would support supervised learning and enhance unsupervised models, providing a solid foundation for anomaly detection while leveraging extensive ADL data.

Modern fall detection datasets, such as UP-Fall [55] and MUVIM [39], offer multiple sensor modalities, but their broad scope often dilutes the focus on optimizing fall detection, leading to inconsistent results. Despite the acknowledged need for personalized systems, existing datasets fail to explore the impact of individual customization on detection accuracy. A key limitation in the field is the absence of a universal benchmark dataset, which hampers the ability to fairly compare models across studies [15, 55, 93]. Privacy and ethical concerns further restrict the creation of real-world datasets, forcing reliance on simulated falls that may not generalize to uncontrolled environments. Although some studies report high accuracy, these results often fail to translate to practical applications due to environmental and participant variability [33].

2.3.2 Lack of a Benchmark Dataset and Challenges in Fair Comparisons

A recurring challenge in fall detection research is the absence of a universally accepted benchmark dataset. This gap prevents fair and consistent comparisons of FDS, stalling progress in the field [15, 93]. Privacy concerns and ethical considerations limit the collection of real-world data, leaving researchers reliant on simulated falls that may not fully capture real-world dynamics [93, 94]. Existing datasets are often specialized to specific environments, further restricting their generalizability [39].

In an ideal scenario, datasets would include genuine falls captured in diverse, real-world settings. However, privacy concerns make such data scarce. Instead, simulated datasets are commonly used, but these often fail to reflect real-world complexities, particularly in uncontrolled environments. The quality of a dataset is critical to determining whether model performance improvements translate to practical outcomes. For instance, while some studies report near-perfect results in lab settings, such as over 99% accuracy and sensitivity [64], these metrics rarely hold up in real-world scenarios [33]. Vision-based solutions, in particular, struggle in uncontrolled environments and low-light conditions, limiting their applicability [81, 95]. Hence, lack of a standardized dataset also

hinders unsupervised approaches, where unexpected actions not well-represented in training data may be misclassified as falls [11]. Regardless of the technique—supervised or unsupervised—a robust dataset with genuine falls is essential for accurate validation and model refinement.

To advance fall detection technology, a comprehensive benchmark dataset is necessary. Such a dataset should include realistic fall dynamics, environmental diversity, and privacy-preserving methods, enabling fair and consistent comparisons across models. The introduction of a benchmark dataset, as detailed in Chapter 4, would drive the development of more accurate models and facilitate the adoption of FDS in real-world applications.

2.4 Fall Detection Solutions

This section reviews fall detection solutions, focusing on approaches relevant to this work, those that inspired it, and those employing similar methods.

2.4.1 Overview

Fall detection solutions (FDS) trace back to Personal Emergency Response Systems (PERS), where users manually pressed a button in emergencies. However, adherence was low, with about 80% of older adults not using the alarm after a fall [18]. This limitation spurred the development of automatic FDS to eliminate reliance on manual activation.

Thermal imaging was explored early in 2004 when Sixsmith *et al.* [96] demonstrated a low-resolution thermal sensor array (16×16) for fall detection. Despite this, wearable devices dominated research for a decade, primarily using accelerometers and gyroscopes with threshold-based or traditional machine learning approaches [15]. Visual-based solutions gained prominence in 2011 with Kinect depth cameras using centroid height and velocity for thresholding. These approaches grew in popularity as deep learning (DL) surpassed traditional methods in fall detection tasks.

Thermal imaging initially had limited application in FDS, primarily for motion detection rather than fall classification [72]. Early systems, like Pyroelectric Infrared (PIR) sensors, provided bi-

nary motion detection based on heat changes [84, 97], but their low resolution and sensitivity to environmental temperature changes limited their utility [98]. For example, in the UpFall solution [55], thermal sensors served only as presence detectors, leading to poor and unreliable fall detection [55]. The absence of dedicated thermal fall detection datasets further impeded progress [71].

Thermal imaging remains underutilized in FDS research. A 2024 review by Gaya *et al.* [71] found that only 6 out of 168 studies utilized infrared data, compared to 132 using RGB cameras and 30 using depth cameras. This disparity stems from historically low-resolution and costly thermal sensors. To address these limitations, thermal sensors are often paired with other modalities, improving performance but increasing system complexity and cost [98]. Additionally, environmental heat sources, such as laptops or heaters, can cause false detections [98].

Recent advancements in thermal sensor technology, such as affordable, higher-resolution devices like the Calumino Thermal Sensor Evaluation Kit v3.1 (CTS-EVK), mitigate these issues. These sensors support robust datasets like the TF-66 dataset introduced in Chapter 4, which includes built-in person detection to reduce false positives caused by environmental heat sources [8].

This thesis addresses a critical gap by demonstrating that modern thermal sensors are suitable for DL applications and cost-effective for large-scale deployment. Unlike earlier systems that struggled with performance, privacy, and cost trade-offs, this work utilizes thermal imaging to develop privacy-preserving, scalable, and effective fall detection solutions.

The following sections review key literature foundational to FDS. Figure 2.3 outlines the structure of these discussions. Traditional machine learning approaches are reviewed to establish a foundation, followed by emerging DL approaches categorized by system modality:

- **Wearable solutions**: Highlighting trends and limitations, emphasizing the shift away from wearable devices.
- **Ambient solutions**: Examining their emergence and limitations.
- **Vision-based solutions**: The primary focus of this thesis, divided into thermal and non-thermal supervised models, and unsupervised models.

- **Ensemble solutions**: Advanced methods integrating multiple modalities, which inspired the ensemble model in Chapter 3.

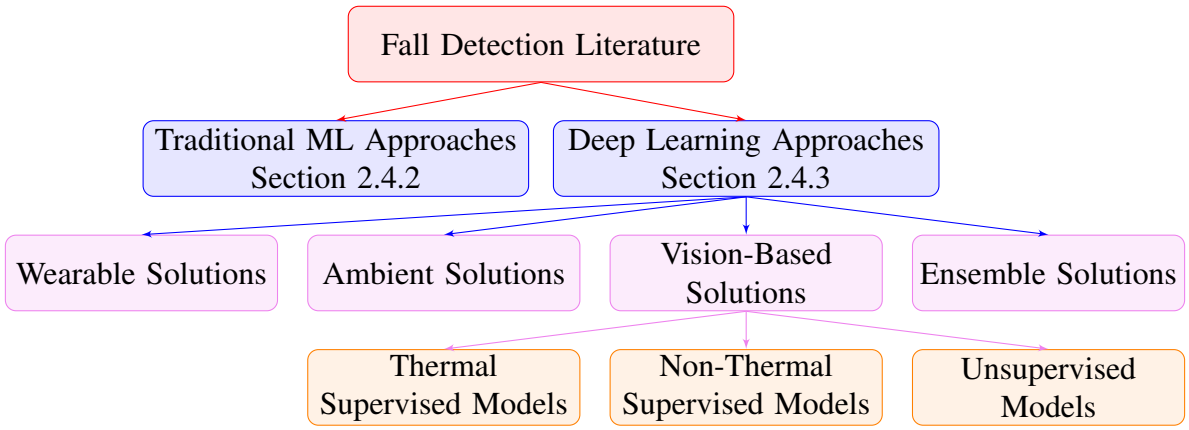


Figure 2.3: Overview of the thematic literature review of fall detection solutions.

2.4.2 Traditional Machine Learning Approaches

Traditional machine learning methods rely on manually extracted features fed into classifiers such as random forests (RF), support vector machines (SVM), decision trees (DT), or logistic regression (LR). Unlike deep learning (DL), these models depend on handcrafted features, such as bounding box information for skeletal positioning, to detect falls based on body posture changes [11, 41, 71, 99]. While effective at the time, these approaches have declined with the rise of DL, which enables automatic feature extraction and reduces reliance on domain expertise. Despite their limitations, reviewing these foundational works is crucial for understanding historical trends in fall detection and creative solutions addressing early challenges. Table 2.2 summarizes key studies employing traditional machine learning methods, highlighting their methodologies, limitations, and performance. A common limitation across all traditional systems is the lack of automatic feature extraction, which ties system performance to developer expertise. This shared limitation is omitted from individual explanations for brevity.

Starting in 2013, researchers explored fall detection using smartphones, which are commonly carried by individuals. Bai *et al.* [12] developed a system leveraging a smartphone's built-in accelerometer to detect falls and determine their direction. The device, worn at the waist near the

center of gravity, achieved 94% accuracy on 50 test samples, distinguishing falls from daily activities. However, this approach was limited by the small dataset size and impracticality of consistently wearing the device at the waist, as many individuals are unwilling or unable to carry it reliably. Subsequent research into smartphone-based fall detection struggled with high false alarm rates caused by constant phone movement, preventing widespread adoption.

Table 2.2: A summary of traditional ML approaches for fall detection
Note: ☒ = Sensitivity, ⋆ = ROC-AUC, △ = Accuracy

Ref.	Methodology	Limitations	Dataset	Performance
[12] 2014	Used an SVM model to identify falls using accelerometers from a smartphone.	Small, private dataset limits comparability. Strict device placement. Abrupt motions incorrectly classified.	Private	94% △
[35] 2020	Evaluates two hearing instrument-based FDS (HIFDS) compared to a pendant using decision trees. (H1 = HIFDS1, H2 = HIFDS2).	Tested with a small participant group of young adults. Connectivity for hearing instruments causes concern for real-world applications.	Private	H1: 92.1% ☒, H2: 80% ☒, Pendant: 82.5% ☒
[100] 2022	Used physiological readings (e.g., blood pressure) with models such as SVM, Naïve Bayes, KNN, etc. A voting ensemble outperformed all models.	Capturing real-time physiological readings is difficult for real-world fall detection. The dataset is no longer public.	Fall Detection Data	Voting Ensemble - 90.9% ⋆
[101] 2022	Used neuro-fuzzy models and wearable sensor data. A majority vote ensemble performed best.	Lack of DL produces poor results. Wearable sensors presents adherence issues.	MobiFall [102]	87.9% △
[23] 2022	Used the Apple Watch Series 5 for wheelchair fall detection using heuristics, showing how watch solutions are poor.	Heuristics require proper thresholding, which may have been a limiting factor.	Private	4.7% ☒

To overcome adherence challenges with wearable fall detection devices, Burwinkel *et al.* [35] proposed embedding fall detection systems into hearing aids, which seniors are more likely to wear consistently. Two hearing aid-based systems with IMU sensors for acceleration and gyroscope data were evaluated using decision tree classifiers. One system achieved a sensitivity of 92.1%, outperforming a fall detection pendant (82.5%) and another hearing aid system (Livio AI B, 80%). However, the small testing group excluded seniors, and connectivity issues limited real-time implementation.

In 2022, Roy *et al.* [100] proposed using physiological data such as blood circulation, heart rate, and blood pressure for fall detection. Individual models, including Logistic Regression, Decision Trees, SVM, Naïve Bayes, and KNN, were combined with a voting ensemble, achieving superior

ROC-AUC performance. Despite its promise, this method is impractical for real-time use due to the need for continuous physiological data collection.

Kordnoori *et al.* [101] explored neuro-fuzzy models for wearable gyroscope and accelerometer signal classification, combining outputs via weighted majority voting. While the ensemble model outperformed individual classifiers, its best performance of 87.9% remains insufficient for critical scenarios like fall detection, underscoring the limitations of neuro-fuzzy systems in this field.

Heuristic thresholding methods were also investigated, using acceleration thresholds to classify falls. Abou *et al.* [23] evaluated the Apple Watch Series 5 for wheelchair fall detection, finding it detected only 4.7% of falls. This poor performance resulted from wrist-based accelerometry fluctuations due to arm movement. Conservative tuning to minimize false alarms further reduced the system's ability to detect critical falls.

2.4.3 Deep Learning Fall Detection Methods

Deep learning (DL) is becoming the dominant approach in fall detection literature due to its capability to create predictive models through automatic feature extraction, provided sufficient data. This section first briefly discusses wearable sensors before exploring ambient methods. The discussion then focuses on related work most relevant to this research—thermal fall detection. Lastly, ensemble methods are reviewed, providing inspiration for the models introduced in Chapter 3.

Wearable Sensor-based Deep Learning Models

While many wearable sensor solutions rely on traditional approaches, DL has also been explored in wearable-based fall detection. Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, model temporal dependencies in sensor data. Despite adherence issues making wearable solutions suboptimal, the following studies offer valuable insights (see Table 2.3).

In 2019, Santos *et al.* [103] proposed a CNN-based fall detection system processing tri-axial accelerometer data from a pendant device. Tested on three public datasets with data augmentation to address class imbalance, the model achieved high sensitivity and specificity (99.72% and 100%

on URFD). However, without augmentation, performance dropped to 83.3%, indicating overfitting risks and limited generalizability due to small, non-diverse datasets.

This underscores a common issue: systems optimized in lab settings often underperform in real-world environments. Notably, CNN architectures consistently outperform LSTM architectures, establishing CNNs as the leading supervised method in fall detection.

In 2023, Yu *et al.* [105] developed a low-power wearable fall detection system using a Tiny Convolutional Neural Network (TinyCNN) to process 3-axis acceleration and angular velocity data. Trained on KFall and SisFall datasets, it achieved over 98% sensitivity and specificity, demonstrating potential for real-time applications. However, it does not address adherence issues, and power consumption remains a challenge, requiring significant power for on-device DL models or continuous data transmission.

Haque *et al.* [41] in 2024 compared LSTM and Transformer architectures for fall detection using smartwatch accelerometer data. While CNN-LSTM models had better offline evaluations (87.6% F1-Score), Transformers showed superior real-world generalizability. The modest F1-Score underscores the insufficiency of wristwatch-based solutions for reliable fall detection.

Although wearable sensors with DL offer robust fall detection capabilities, challenges like consistent device placement, user adherence, and battery life limit widespread adoption. These

Table 2.3: A summary of the key DL-based approaches that employ a wearable sensor modality
Note: ☒ = Sensitivity, ⋆ = ROC-AUC, △ = Accuracy, ⋄ = F1-Score

Ref.	Methodology	Limitations	Dataset	Performance
[103] 2019	Uses shallow CNNs with wearable accelerometer data, leveraging data augmentation (DA) to balance classes, outperforming LSTM solutions.	System is reliant on the DA process, producing great results in lab experiments that will not translate to real-world falls.	URFD [104] (accelerometer data only)	W/t DA: 99.72% ☒, W/o DA: 83.33% ☒
[105] 2023	Uses a tiny CNN which decouples temporal and spatial feature extraction to detect falls based on wearable inertial sensors.	Reliance on wearable systems. Highly battery-consuming.	KFall [106], SisFall [107]	KFall - 99.83% ☒, SisFall - 98.29% ☒
[41] 2024	Compares LSTM and Transformer (TF) architectures both offline (OF) and online (ON) using data collected from smartwatches.	Private real-world test with only 3 people, producing inverse results. Wrist placement causes more false alarms.	SmartFall [108], UniMib [109], KFall [106]	OF: TF - 82.6% ⋄, LSTM - 87.6% ⋄. ON: TF - 77.1% ⋄, LSTM - 70.2% ⋄. All on SmartFall

constraints, along with potential user discomfort and limited generalizability, reduce their viability. Nonetheless, architectural innovations from wearable research inspire the development of advanced, privacy-preserving, vision-based FDS.

Ambient Sensor-based Deep Learning Models

Ambient sensors, or environmental sensors, provide privacy-preserving and non-intrusive solutions for fall detection. However, vibration- and pressure-based approaches often experience high false alarm rates [52]. Recent advancements in radar and WiFi-based technologies have shown promise in mitigating these issues through advanced signal processing and DL techniques. These technologies have gained significant attention in the fall detection community, with key studies summarized in Table 2.4.

One prominent modality for radar-based solutions is mmWave radar. Kittiyanpunya *et al.* [110] developed a fall detection system utilizing LSTM networks to classify falls based on 1D point clouds in the z-axis (floor-to-ceiling) and Doppler velocity data. The system achieved 99.5% accuracy across five controlled environments. However, its dataset consisted of only ten participants performing simulated falls, limiting its generalizability. Additionally, the lack of publicly available datasets and reliance on controlled experimental conditions hinder its applicability in dynamic, real-world settings.

Table 2.4: A summary of the key DL-based models that use ambient sensor modalities
Note: △ = Accuracy

Ref.	Methodology	Limitations	Dataset	Performance
[110] 2023	Utilizes z-axis point clouds and Doppler velocity from mmWave radar as input to an LSTM.	Private dataset limits comparability. Reliance on controlled environments limits performance in real-world settings.	Private	99.5% △
[16] 2023	Employs UWB radar with adaptive channel selection and CNNs.	Private dataset limits comparability. Only 3 fall types tested. All samples recorded 1.5 meters away.	Private	95.7% △
[60] 2023	Implements a fine-tuned EfficientNet using CSI variations from WiFi data.	Private dataset limits comparability. Many false positives without additional samples outside the line of sight.	Private	96.8% △

In 2023, Wang *et al.* [16] proposed a fall detection system using ultra-wideband radar. Through adaptive channel selection and lightweight CNN models, the system achieved 95.7% detection ac-

curacy across three fall types: stand-to-fall, bow-to-fall, and squat-to-fall. However, the small sample size (nine participants) and limited fall diversity restrict its applicability to real-world environments. Moreover, recordings were confined to within 1.5 meters of the radar, further limiting generalizability.

WiFi signals have also emerged as a promising modality for fall detection. Chu *et al.* [60] developed a WiFi-based system leveraging Channel State Information (CSI) to detect signal amplitude and phase variations caused by human motion. Using a deep learning classifier, the system achieved over 96% accuracy in four indoor environments. Despite these results, the dataset included only younger participants (ages 24–43), excluding elderly individuals whose fall dynamics differ significantly. Line-of-sight limitations and interference from other devices and networks further reduced performance in real-world settings.

While these studies showcase potential, their reliance on small, private datasets limits comparability and real-world applicability. High accuracy achieved under controlled conditions may not extend to broader scenarios. Addressing these challenges requires large, diverse datasets encompassing varied populations, environments, and fall scenarios.

Beyond dataset limitations, ambient sensors face practical challenges such as high false alarm rates, difficulty identifying individuals in shared environments (e.g., retirement facilities), and sensitivity to environmental factors. WiFi-based systems are particularly affected by noise, furniture placement, and crowded frequency bands, while radar systems require precise calibration and antenna positioning.

Despite these challenges, ambient sensors remain promising for privacy-preserving, non-contact fall detection. Advancing their generalizability, robustness, and scalability is essential for real-world adoption.

Vision-based Deep Learning Models

These solutions leverage image-based data for fall detection and are a prominent focus in the research community. A 2022 review of vision-based human FDS employing DL revealed that over 80% of systems using a single model type relied on convolutional neural networks (CNNs) [7].

This positions CNNs as the traditionally most popular solution for fall detection problems. However, recent advancements have increasingly highlighted the potential of AEs, both in supervised and unsupervised implementations.

Thermal Data Supervised Models: Supervised thermal FDS frequently rely on low-resolution sensors, which can limit their ability to capture detailed body movements and subtle pre-fall indicators. This constraint reduces overall system performance and may lead to missed falls in real-time applications, posing significant risks for practical deployments. Despite these challenges, thermal data remains a promising modality due to its inherent privacy-preserving qualities. Key research in this domain is summarized in Table 2.5 and discussed in detail below. These studies highlight advancements in supervised DL approaches for thermal vision-based fall detection and address the trade-offs between resolution, performance, and real-world applicability.

Starting in 2018, when affordable thermal sensors were scarce, Taramasco *et al.* [111] used low-resolution (1×8) thermal sensors with RNN subtypes (Bi-LSTM, GRU, and LSTM) combined with convolutional layers. Bi-LSTM achieved the best performance, while GRU provided greater efficiency, highlighting the trade-off between accuracy and complexity in fall detection models.

Rafferty *et al.* [27] later utilized ceiling-mounted 32×32 thermal sensors to develop three methods: a logical approach, a CNN-based model, and a composite analysis. The logical approach, detecting rapid blob expansion over three frames, achieved 76% accuracy in simulated settings. The CNN model improved accuracy to 80% but was limited by a small dataset of 119 non-fall and 211 fall images. The composite analysis method, capturing temporal aspects with multi-frame composites, performed poorly at 35% accuracy.

In 2020, Tateno *et al.* [62] achieved 94% accuracy using LSTM models with 32×24 infrared sensors to classify activities such as falling, sitting, and standing. CNN models, which achieved 91.5%, were hindered by manual feature extraction and the lack of combined spatial-temporal analysis, resulting in false positives such as individuals lying on the floor.

Naser *et al.* [8] utilized 32×24 thermal arrays and optical flow features classified with LSTM and Bi-LSTM models. A human-in-the-loop confirmation reduced false positives, achieving 99.7%

Table 2.5: A summary of the key supervised DL works using a thermal vision-based modality
Note: ☒ = Sensitivity, △ = Accuracy

Ref.	Methodology	Limitations	Dataset	Performance
[111] 2018	Employed 1×8 thermal sensors with RNN subtypes. Bi-LSTM (Bi) achieved best performance, with GRU (G) offering greater efficiency.	Resolution too low to be relied upon. Sensor grids installed low on wall wall leading to high false alarm rates.	Private	LSTM: 91% △, G: 87.5% △, Bi: 93% △
[27] 2019	Used ceiling-mounted, low-resolution 32×32 thermal sensors for 3 methods; logical, CNN, and a composite of the two. CNN performed best.	Tracking blob expansion over 3 frames doesn't provide enough context for the logical method, the resolution and number (330) of samples for the CNN are small.	Private	Logical: 76% △, CNN: 80% △, Composite: 35% △
[62] 2020	Extract 6 features from a 32×24 thermal sensor and manually pass into a CNN and LSTM, with LSTM performing best.	Private dataset limits comparability. Low resolution sensor. Lack of temporal feature processing. Manual features extracted are likely suboptimal.	Private	CNN: 91.5% △, LSTM: 94% △
[8] 2022	Used low-resolution (32 × 24) thermal sensors to demonstrate how integrating optical flow can increase performance on LSTM and Bi-LSTM on average by 22.7%.	Wall-mounted sensors increase occlusions. Low resolution sensors are unreliable. Private dataset limits comparability. Small dataset of 226 falls limits system generalizability.	Private	99.7% △
[31] 2022	Used low-resolution thermal cameras (80×60) to classify points as either on the ground or above the ground using a SVM and a DenseNet.	The use of multiple sensors introduces calibration issues. Wall-mounted sensors introduces occlusions. Could misclassify when cleaning on ground.	Private	97.6% △, 90.94% ☒
[84] 2023	Used a PIR and a low-resolution (32×32) thermal sensor into a fully connected 3-layer neural network.	Small, private dataset limits comparability. Wall-mounted sensors increase occlusions. Only works in the bathroom.	Private	92.81% △
[78] 2023	Used two 24×32 low-resolution thermal sensors simultaneously into various models, with CNN outperforming all models.	Private dataset limits comparability. Using single frames limits critical temporal features. The system was also sensitive to ambient temperatures.	Private	97.9% △

accuracy with Bi-LSTM. However, reliance on a small, homogeneous dataset and wall-mounted sensor placement caused occlusion issues, limiting generalizability.

As thermal camera resolutions improved, Zoetgnande *et al.* [31] in 2022 employed a stereo setup with 80×60 thermal cameras and a DenseNet-inspired deep network, achieving 97.6% accuracy. Challenges included occlusions, calibration complexity, and limited real-world validation, which hindered continuous real-time use.

He *et al.* [84] focused on bathroom environments with a 32×32 thermal sensor and PIR motion detector, achieving 93% accuracy with a simple three-layer fully connected neural network. Low resolution, occlusion from wall-mounted placement, and a small dataset limited performance.

In 2023, Rezaei *et al.* [78] combined two 24×32 thermal sensors to classify activities with CNNs, achieving 97.9% accuracy. However, reliance on single-frame analysis limited the system's ability to address spatial-temporal aspects of falls, raising the risk of false alarms in real-time applications.

These studies highlight thermal cameras' potential for privacy-preserving fall detection, especially in varying lighting conditions. However, shared limitations—such as reliance on private datasets, low-resolution sensors, and restricted generalizability—underscore the need for higher-resolution thermal sensors, like those in the TF-66 dataset introduced in Chapter 4, to advance the field.

Non-Thermal Data Supervised Models: Researchers frequently utilize RGB cameras for vision-based fall detection, but these systems face significant challenges. Privacy concerns arise from capturing RGB video, which can reveal sensitive information. Additionally, their reduced effectiveness in low-light or dynamic environments limits robustness and applicability. Many models are trained on small, simulated datasets that lack demographic diversity, fall types, and environmental variability, resulting in poor generalizability.

The lack of real-world testing further exacerbates these issues, as these systems often fail to account for the slower, more cautious fall dynamics typical of elderly individuals compared to the simulated scenarios used for training. Table 2.6 summarizes key studies on non-thermal supervised fall detection models, highlighting their performance and limitations.

Non-thermal supervised models are generally more computationally advanced than their thermal counterparts, as evidenced by key research. In 2017, Núñez-Marcos *et al.* [30] proposed a system using optical flow images and a modified VGG-16 network, achieving high sensitivity (100%, 99%, and 93.47%) on the URFD, Multicam, and FDD datasets, respectively. The system focused exclusively on motion patterns, ensuring invariance to environmental features like lighting and background. It employed a multi-stage training strategy: pretraining on ImageNet for feature extraction, fine-tuning on the UCF101 action-recognition dataset for motion recognition, and

further fine-tuning on fall detection datasets. While effective, this approach shares common RGB camera limitations, including ineffectiveness in dark environments and privacy concerns.

In 2019, Brieva *et al.* [10] extended fall detection research by using optical flow from RGB videos as input to a CNN model, incorporating a majority voting approach to reduce false positives. While this method improved performance, it relied on private datasets, limiting comparability.

To address privacy concerns in RGB-based solutions, Kong *et al.* [81] introduced a three-stream CNN integrating human silhouettes, motion history images, and dynamic images from surveillance video data. The system achieved state-of-the-art performance, with 100% sensitivity and 99.3% specificity on the MCF and UR Fall datasets. However, silhouette extraction relied on static backgrounds, reducing reliability in dynamic environments, and real-time processing of silhouettes and motion images was computationally intensive.

Table 2.6: A summary of vision-based DL models w/t non-thermal modality
Note: ☒ = Sensitivity, △ = Accuracy

Ref.	Methodology	Limitations	Dataset	Performance
[30] 2017	Uses optical flow extracted from RGB as input to a VGG-16 network pretrained on ImageNet and tuned on UCF101 for motion recognition.	The datasets used lack diversity in environmental scenarios, demographics, and fall types, limiting the model's ability to generalize to real-world applications.	URFD [104], MultiCam [112], Le2i [113]	URFD: 100% ☒, MultiCam: 99% ☒, Le2i: 93.47% ☒
[10] 2019	Used a basic CNN with optical flow from RGB images. Majority voting increased performance.	Private dataset limits comparability. Only 5 frames (300ms of real-time input) are used for analysis.	Private	92.78% △
[81] 2019	Proposed a three-stream CNN, inputs being human silhoutettes, motion history images, and dynamic images extracted from RGB inputs.	Only works in well-lit environments, so not at night. The effectiveness of silhouette extraction relies on static backgrounds.	MultiCam [112] and URFD [104]	MultiCam: 98.2% ☒, UR Fall: 100% ☒
[64] 2022	Proposes a 4-stream 3D CNN model in which each stream focuses on phases of a fall; standing, falling, fallen, rest.	Dataset used has poor diversity. Splitting falls into phases won't always capture real-world falls, limiting generalizability.	Le2i [113]	99.03% △, 99.00% ☒
[57] 2023	Integrating both RGB and optical flow data into a weakly supervised learning approach using the I3D network.	Only works in well-lit environments, so not at night. The introduced RFDS (R) dataset only has 240 videos.	URFD(U)[104], Le2i(L)[113], RFDS [57]	U: 98.4% △, L: 99.0% △, R: 97.2% △.
[34] 2024	Introduces an enhanced YOLOv7-based model trained on RGB images, with inclusion of attention mechanisms to increase performance.	Private dataset limits comparability and lacks diversity in participant demographics and real-world variability. Only works in well-lit environments.	Private	89.8% ☒
[61] 2024	Combines a lightweight 3D CNN (with channel- and spatial-wise attention mechanisms) with ConvLSTM networks on RGB data.	Using single frames limits critical temporal features. The system was also sensitive to ambient temperatures.	URFD [104], MultiCam [112]	URFD: 98.07% ☒, MCFD: 96.88% ☒

In 2022, Alanazi *et al.* [64] proposed a multi-stream 3D CNN architecture (4S-3DCNN) that segmented falls into four phases: standing, falling, fallen, and at rest. Each phase was processed in separate branches to learn spatial and temporal features. Despite achieving 99.03% accuracy on the Le2i dataset, the approach required extensive preprocessing and assumed a standard fall progression, making it less effective for atypical scenarios like gradual or partial falls.

In 2023, Wu *et al.* [57] revisited optical flow solutions with a dual-modal system combining RGB and optical flow data within a weakly supervised framework. Using multiple instance learning, the system reduced labeling requirements and achieved state-of-the-art accuracy on URFD, Le2i, and RFDS datasets by combining RGB spatial features and optical flow temporal features through late fusion. However, its reliance on RGB data limited performance in low-light conditions and raised privacy concerns. Additionally, the small RFDS dataset (240 videos) restricted generalizability.

In 2024, attention mechanisms became a focus in fall detection systems (FDS). Zhao *et al.* [34] introduced YOLO-Fall, a lightweight YOLOv7-based model with attention modules to reduce computational complexity. It achieved 89.8% sensitivity on a custom dataset, but over 10% of falls went undetected, highlighting the need for improvement. Su *et al.* [61] proposed combining lightweight 3D CNN and ConvLSTM architectures with channel- and spatial-wise attention modules to model long-term spatial-temporal dependencies. This method achieved 98.07% sensitivity on URFD and 96.88% on MCFD datasets.

While these systems demonstrate strong performance in controlled environments, they share significant limitations. Most are not fully privacy-preserving, rely on specific environmental setups and lighting, and lack generalizability to real-world conditions. To address privacy concerns, some systems focus on extracting human silhouettes or skeletal information rather than processing raw video data. A summary of these systems is provided in Table 2.7.

An early adopter of silhouette-based fall detection methodologies, Asif *et al.* [22], proposed FallNet, a framework leveraging synthetic RGB data for privacy-preserving fall detection. Using human pose and segmentation data with a multi-modal CNN, the system achieved F1-Scores of 0.8708 on the MultiCam dataset and 0.9244 on the Le2i dataset. FallNet was trained entirely

Table 2.7: A summary of key supervised DL models that address privacy concerns associated with RGB data by leveraging alternative approaches
Note: △ = Accuracy, ⋄ = F1-Score

Ref.	Methodology	Limitations	Dataset	Performance
[22] 2020	Proposed framework leverages synthetic RGB data with human pose and segmentation using a multi-modal CNN, achieving good generalization to real-world datasets.	The model was trained on synthetic data and then tested on the listed datasets. While innovative, this limits applicability in challenging environments with occlusions or varied camera angles.	MultiCam [112], Le2i [113]	MultiCam: 87.08% ⋄, Le2i: 92.44% ⋄
[94] 2020	Extended the previous work by focusing on image-based fall detection, integrating a 3D pose estimation module into the system.	Performance decreased while computational complexity increased.	MultiCam [112], Le2i [113]	MultiCam: 84.53% ⋄, Le2i: 89.91% ⋄
[59] 2023	Obtain human silhouette using a pixel-level classification model, which are then fed into a ConvLSTM.	Preprocessing step increases complexity. Use of RGB dataset limits usability in real-world systems.	URFD [104] and UPFall [55]	URFD: 97.68% ⋄, UR Fall: 100% ⋄
[28] 2023	Proposed a skeleton-based system using an advanced DL model which uses spatiotemporal graphs and attention mechanisms.	Slow processing (6.787 seconds per sample) and reliance on AlphaPose for skeleton extraction limit robustness in complex, occluded real-world rooms.	URFD(U)[104], Le2i(L)[113], FDD(F)[114]	U: 99.7% △, L: 99.93% △, F: 99.12% △

on synthetic data, generated with 3D humanoid models in Blender and simulated with MoCap data. While innovative, reliance on synthetic data limited its applicability to existing recorded fall detection datasets.

Later in 2020, Asif *et al.* [94] extended this work by incorporating 3D pose estimation to enhance resilience against occluded joints. However, this addition reduced F1-Scores by approximately 2.5% on each dataset, demonstrating a common issue in fall detection research: models optimized for training datasets often lack scalability and transferability to real-world scenarios. In this case, attempts to improve robustness decreased overall performance, underscoring the limitations of lab-based optimizations.

Lab-focused approaches continued with Mobsite *et al.* [59], who developed a silhouette-based system combining MSSkip for pixel-level classification with ConvLSTM for sequence learning. The system achieved impressive F1-Scores of 97.68% on the UP Fall dataset and 100% on the UR Fall dataset. However, the computational demands of pixel-level classification make it impractical

for real-time use. Additionally, while silhouettes enhance privacy, RGB video capture remains a prerequisite, which raises privacy concerns and discomfort among seniors.

Egawa *et al.* [28] proposed a skeleton-based system using a graph-based spatial-temporal convolutional and attention neural network. Input images were converted into skeleton representations of human joints, with motion extracted from differences between consecutive frames. These frames were constructed into graphs, allowing spatial-temporal convolutional neural networks to extract contextual relationships. An attention module refined features channel-wise before classification. Despite achieving over 99% accuracy on three benchmark datasets, the system's inference time of nearly 6.8 seconds per sample makes it unsuitable for real-time applications.

These innovative approaches share several common limitations:

- **Privacy Concerns**: RGB cameras, though enabling pose estimation and segmentation, require raw frame capture, raising significant privacy risks.
- **Computational Overhead**: Pose estimation, segmentation, and pixel-level classification impose high computational demands, hindering real-time performance.
- **Dataset Limitations**: Validation often relies on small, controlled datasets that lack diversity in demographics, environments, and fall scenarios, limiting generalizability.

Unsupervised Models: Unsupervised fall detection systems (FDS) treat falls as anomalies, identifying unusual events based on reconstruction errors or predictive discrepancies. These methods, typically employing autoencoders (AEs), adversarial networks, or predictive frameworks, are trained exclusively on normal daily activities to minimize reconstruction or prediction errors. Their popularity stems from the scarcity of large, comprehensive datasets simulating real-world fall scenarios [115].

Despite their innovative designs, unsupervised FDS face several limitations:

- **Dataset Limitations**: Small, controlled datasets with limited demographic and environmental diversity restrict generalizability. This lack of diversity increases the likelihood of false positives when encountering underrepresented scenarios.

- **Anomaly Misclassification**: Anomaly detection frameworks risk misclassifying unrelated anomalies as falls. Since all unknown actions are classified as falls, training datasets must be highly diverse, covering all possible actions the system might encounter during deployment.
- **Threshold Sensitivity**: Many solutions rely on thresholds to determine if an AE's reconstruction error is "high enough" to indicate an anomaly. This dependency makes systems highly sensitive to individual differences and motion variability, reducing generalizability to external scenarios.
- **Input Modality Concerns**: Reliance on RGB datasets adds challenges, as these datasets must effectively represent diverse actions across various environments, angles, and actors. Achieving this diversity in practice is difficult, further limiting real-world effectiveness [105].

These limitations are common across key unsupervised fall detection studies, summarized in Table 2.8. The next section discusses the most relevant unsupervised thermal fall detection solutions.

In 2017, Chong and Tay [119] laid the foundation for unsupervised video anomaly detection by introducing a spatiotemporal autoencoder for anomaly detection in video inputs. Building on this, Seyfioğlu *et al.* [120] extended this approach in 2018 to classify various actions, comparing AEs, CNNs, ConvAEs, and SVMs. AEs performed best, though the system struggled to differentiate between general falling, falling off a chair, and sitting. This study utilized radar-based micro-Doppler signatures rather than vision-based modalities.

Inspired by these advancements, Nogas *et al.* [116] introduced a vision-based Convolutional LSTM Autoencoder (ConvLSTM-AE) for fall detection. By combining convolutional layers for spatial encoding/decoding with LSTM layers for temporal patterns, the model outperformed standard AEs, CAEs, and DAEs, achieving an AUC of 0.83 on the TSF dataset. However, as a proof-of-concept, its performance was insufficient for real-world deployment.

In 2020, Nogas *et al.* [76] improved upon this work with a 3D Convolutional Autoencoder (3DCAE), achieving a 97% ROC-AUC on the TSF dataset. This approach introduced anomaly scoring based on contiguous frame windows, enhancing temporal context. However, using only

Table 2.8: A summary of the key unsupervised DL works with vision-based modality
Note: ☒ = Sensitivity, ⋆ = ROC-AUC

Ref.	Methodology	Limitations	Dataset	Results
[116] 2018	Used a ConvLSTM-AE on visual-enhanced thermal ADL images, identifying high reconstruction scores of data as anomalies indicating falls.	The system had mediocre results, not being reliable enough for real-world implementation.	TSF [2]	83.0% ⋆
[76] 2020	Improved the previous work by using a 3D Convolutional Autoencoder instead of the ConvLSTM-AE.	Small input window size misses vital temporal contextual information.	TSF(T)[2], URFD(U)[104], SDU(S)[117]	T: 97.0% ⋆, U: 86.0% ⋆, S: 95.0% ⋆
[118] 2020	Uses a Spatiotemporal Residual Autoencoder, using convolutional layers for spatial features, ConvLSTMs for temporal features, residual connections for efficiency.	Only use 8 frames per sample. Dataset is 12fps, meaning only 0.67 seconds are captured each classification, missing vital temporal contextual information.	TSF [2]	97% ⋆
[75] 2021	Employs a spatio-temporal adversarial framework consisting of a 3D convolutional autoencoder for reconstructing sequences of ADL and a 3D CNN as a discriminator.	System had worse results than previous solutions for TSF. The adversarial framework's reliance on reconstruction and discrimination adds computational overhead.	TSF(T)[2], URFD(U)[104], SDU(S)[117]	T: 95.0% ⋆, U: 91.0% ⋆, S: 91.0% ⋆
[70] 2021	Introduced a dual-channel adversarial network processing thermal frames and optical flow data. They also added region of interest extraction.	System performed worse than the previous adversarial model and relies on accurate ROI extraction, which can fail with occlusion or tracking errors.	TSF [2]	93.0% ⋆
[115] 2023	Unlike reconstruction-based AEs, an attention-based U-Net predicts future video frames, comparing predictions to frames.	The system excelled on simpler datasets but struggled with more complex environments.	URFD(U)[104], MultiCam (M) [112], HQFDS [93]	U: 100% ☒, M: 100% ☒, HQFDS: 68.4% ☒
[11] 2024	Introduces a novel multi-objective loss function, Temporal Shift, and evaluates it using a 3DCAE, an attention U-Net (U), and a multi-modal network.	Performance metrics for 3DCAE and multi-modal network were not disclosed. The loss function requires precise tuning, limiting its generalizability.	MUVIM [39]	U: 92.0% ⋆

8 frames (0.67 seconds at 12 fps) per sample limited contextual understanding of falls. Similarly, Elshwemy *et al.* [118] achieved a 97% ROC-AUC with a Spatiotemporal Residual Autoencoder (SRAE) that integrated ConvLSTM layers and residual connections for improved efficiency.

In 2021, Khan *et al.* [75] adopted spatiotemporal adversarial networks inspired by Lee *et al.* [121]. These networks employed a generator and discriminator, achieving a competitive 95% ROC-AUC on the TSF dataset. However, the adversarial framework introduced significant computational overhead without surpassing simpler models. Mehta *et al.* [70] advanced this work by integrating dual-channel CAEs for reconstructing thermal data and optical flow. Despite adding motion constraints and ROI extraction to reduce noise, the model achieved a lower ROC-AUC of 0.93, constrained by its reliance on accurate ROI extraction.

In 2023, Li *et al.* [115] introduced a frame prediction approach that compared predicted frames to actual ones, leveraging intensity, gradient, and optical flow constraints to improve prediction quality. While achieving perfect performance on URFD and MultiCam datasets, the system classified only two-thirds of falls correctly on the HQFDS dataset, underscoring the need for more robust datasets reflective of real-world conditions. Denkovski *et al.* [11] proposed a Temporal Shift loss function combining reconstruction and prediction losses. Applied to state-of-the-art models such as 3DCAE [76] and attention U-Net [115], it boosted ROC-AUC by up to 20%, reaching 0.92. However, the use of the MUVIM dataset, which combines thermal and visible light data, compromised privacy preservation, and precise hyperparameter tuning limited generalizability.

Unsupervised models demonstrate significant potential for leveraging anomaly detection in privacy-preserving fall detection. However, their dependence on limited datasets, computational complexity, and modality-specific challenges highlight the need for broader validation and optimization to enable practical real-world deployment.

Ensemble-based Models:

Ensemble methods in fall detection systems (FDS) are typically categorized into three approaches:

- **Stacking**: Combines predictions from multiple base models via a meta-model that learns to make final predictions [122].
- **Weighted Average**: Aggregates predictions from base models, assigning weights proportional to their individual performance [123].
- **Class-Based**: Utilizes specialized models or sub-networks tailored to specific classes, enhancing class-specific accuracy [123].

Ensemble methods enhance system performance by integrating additional context through multiple models or specialized approaches, leading to more accurate predictions. Consequently, they have garnered significant attention for improving FDS. This section explores the implementation of ensemble methods in existing fall detection solutions, focusing on their effectiveness in boosting system performance.

Basic majority voting methods are excluded from this discussion, as they are more suitable for smaller, less complex models and are less relevant for real-world deployments involving advanced DL architectures. Instead, this section highlights advanced ensemble techniques that combine multiple modalities or methods into cohesive systems, offering insights into developing robust fall detection solutions.

While ensemble methods are widely used in wearable and RGB-based FDS, their application in thermal imaging remains limited. Comprehensive reviews [7, 73] highlight this gap, with Ogawa and Natio [124] being a notable exception. Their voting ensemble outperformed individual models, demonstrating the potential of ensemble strategies in thermal-based systems.

A summary of key ensemble-based solutions integrating multiple modalities or methods is provided in Table 2.9.

Table 2.9: A summary of the key ensemble DL models, excluding basic voting ensembles
Note: △ = Accuracy, ⋄ = F1-Score

Ref.	Methodology	Limitations	Dataset	Performance
[122] 2018	Compared majority voting and stacking from base-classifiers to classify wearable input data. Stacking using a MLP as the meta-classifier maximized performance.	Small dataset limits generalizability, manual feature extraction limits robustness.	Private	Arm Sensor: 94.88% △, Waist Sensor: 100% △
[32] 2020	Combines pressure sensor, accelerometer, and gyroscope data through independent CNNs, with outputs merged in a fully connected layer for final classification, outperforming individual models.	Ensemble branches are different modalities, increasing complexity and introducing synchronization issues.	Private	99.3% △
[17] 2021	Proposed a stacking ensemble of LSTM networks with varying neuron counts per layer, where a meta-classifier integrated predictions and outperformed individual models.	Small dataset limits generalizability. Wearable solution is suboptimal.	SmartFall [108]	97.8% ⋄
[125] 2023	Integrated a coarse-fine CNN and GRU into a meta-learner, capturing spatial and temporal features from wearable data. The meta-learner outperformed individual models.	Wearable solution is suboptimal. Small dataset used, including samples recorded from 15 participants.	FallAllD [126]	94.26% ⋄

Starting in 2018, Hnoohom *et al.* [122] utilized accelerometer and gyroscope data for fall detection, extracting statistical features such as mean, standard deviation, and signal magnitude area. They proposed a stacking meta-classifier to combine outputs from base classifiers, comparing ensemble methods (majority voting and stacking) with individual models like Decision Trees, Multilayer Perceptrons (MLP), and SVMs. Stacking with MLP as the meta-classifier achieved the

highest accuracy: 94.88% for the arm sensor and 100% for the waist sensor. However, the study was constrained by a small dataset of ten participants aged 18–25 and its reliance on manual feature extraction, reducing adaptability to new activities or configurations.

In 2020, Wang *et al.* [32] combined three input modalities—foot-insole pressure sensors, wearable accelerometers, and wearable gyroscopes—into a CNN ensemble. Each modality was processed by its own CNN, with outputs combined into a feature map, reduced via principal component analysis, and classified through a fully connected layer. This approach achieved an average accuracy of 99.3% with a false positive rate below 0.69%, validated on a private dataset containing 800 falls and 1000 ADL samples from 10 participants. Despite its strong performance, this method increased computational complexity and posed synchronization challenges for real-time applications.

In 2021, Farsi *et al.* [17] proposed an ensemble with branches comprising LSTM models featuring different neuron counts per layer. A stacking meta-classifier integrated predictions, outperforming standalone LSTM models in precision, recall, and F1-Score. While effective in addressing data imbalance and limited training data, the system relied on the small, controlled SmartFall dataset, which limited its generalizability to diverse real-world scenarios.

In 2023, Liu *et al.* [125] developed a coarse-fine CNN combined with a GRU for fall detection using tri-axial data recorded from an inertial measurement unit worn on the neck, waist, and wrist. The coarse branch captured broad spatial features, the fine branch extracted detailed spatial features, and the temporal branch employed GRU layers to model temporal dependencies. Outputs from all branches were concatenated and classified via a fully connected layer. Evaluated on the FallAllD public dataset, the model achieved an F1-Score of 94.26%. Despite showcasing the benefits of ensemble learning, the reliance on wearable data—a suboptimal modality—and a small dataset with only 15 participants limited its scalability to real-world conditions.

These examples illustrate that ensembles employing a meta-learner to combine outputs from multiple models consistently improve performance. However, computational complexity, a key challenge for real-time deployment in residential environments, is rarely addressed in detail. The

inherent complexity of ensemble methods could limit their practicality for FDS requiring continuous operation.

Multi-stream architectures, often confused with ensembles, process different modalities (e.g., radar, thermal) in parallel within a unified network rather than aggregating outputs from distinct models. For instance, Zhou *et al.* [50] combined modalities within a single network, in contrast to Wang *et al.* [32], who treated modalities as branches in an ensemble, each feeding into a dedicated CNN.

The scarcity of ensemble approaches for thermal imaging presents a significant opportunity for innovation. Notably, none of the reviewed studies combine supervised and unsupervised models into a stacking meta-learner. This hybrid approach, which leverages the strengths of both learning paradigms while mitigating their limitations, could enhance performance. Chapter 3 explores this hybrid ensemble approach.

2.5 Chapter Summary

Despite achieving near-perfect performance in research settings, the lack of real-world implementation leaves fall detection literature at a crossroads. Researchers have optimized models extensively, yet a clear disconnect exists between lab results and real-world outcomes. Future work must prioritize comparing fall detection methods on comprehensive datasets that represent the environments and scenarios in which these systems will be deployed. Such datasets will highlight the limitations of proposed systems and identify models that demonstrate true progress in the field, moving beyond lab-specific optimizations to real-world solutions.

The next Chapter proposes a novel architecture combining unsupervised and supervised models via a meta-learner, inspired by ensemble solutions. This hybrid approach aims to leverage the advantages of both learning paradigms while addressing their respective challenges, providing a foundation for robust FDS applicable to real-world environments.

Chapter 3

A Joint Supervised and Unsupervised Fall Detection Model

3.1 Overview

Thermal imaging is a promising, non-invasive technology for fall detection, especially in low-light conditions. However, accurately detecting falls in thermal images poses challenges due to the low resolution and lack of color information in this data. This chapter introduces a novel approach to improving fall detection in thermal imagery through a stacking ensemble of AE and 3D Convolutional Neural Network models, integrated into a meta-neural network trained to classify falls and non-falls.

The proposed system harnesses both supervised and unsupervised learning to mitigate the individual limitations and biases of each model type, presenting a balanced solution for accurate, efficient fall detection. The effectiveness of this approach is demonstrated through ablation studies conducted on the benchmark "Thermal Simulated Fall" dataset, setting a foundation for advancements in this field. The rationale behind combining supervised and unsupervised models lies in their complementary strengths, particularly when working with a limited dataset. Supervised models can struggle with unseen or rare fall scenarios, which may lead to misclassifications due to insufficient fall samples. In contrast, unsupervised models, trained to reconstruct only non-fall

actions, may flag a fall as an anomaly due to high reconstruction error—yet risk producing false positives in real-world usage without extensive non-fall data. Thus, using a meta-model to integrate the strengths of both approaches aims to increase detection accuracy while minimizing false alarms, addressing a primary concern for users who require reliable fall detection without frequent false alerts.

Thus, this Chapter examines the feasibility of this combined approach, assessing whether the integration of supervised and unsupervised models through a meta-model offers a meaningful improvement over individual models. If successful, this approach could direct future research efforts toward hybrid model architectures for robust fall detection in thermal imaging.

3.2 Ensemble Model w/t Supervised & Unsupervised Learning

As discussed in Chapter 2, supervised and unsupervised models each bring unique strengths and limitations to fall detection. Supervised models excel at leveraging labeled data for precise classification but often face challenges with class imbalance and data scarcity, particularly in fall detection where falls are rare. On the other hand, unsupervised models frame falls as anomalies, learning patterns from unlabeled data to detect deviations. However, threshold-based anomaly detection in unsupervised approaches can lead to false positives and limited generalizability in dynamic real-world environments. Given these complementary strengths and weaknesses, hybrid approaches that integrate supervised and unsupervised models hold significant promise. For example, convolutional AEs pre-trained in an unsupervised manner have been shown to enhance classification accuracy when combined with supervised CNNs [120]. This synergy suggests that hybrid ensembles can address key challenges in fall detection, particularly for datasets with diverse scenarios and limited labeled data.

3.2.1 The Case for Ensembles in Fall Detection

Ensemble learning amplifies the benefits of hybrid approaches by combining multiple models to improve robustness, generalization, and performance. Chapter 2 highlighted how stacking en-

sembles and voting-based approaches have consistently outperformed individual models in fall detection studies [127, 128]. By integrating supervised and unsupervised learning through a stacking ensemble, one can leverage the nuanced feature discovery of unsupervised models alongside the discriminative power of supervised learning. This Chapter introduces a novel stacking ensemble framework tailored for fall detection, designed to: improve generalization to diverse, unseen environments, reduce false positives through the robust feature extraction of unsupervised models, and enhance reliability by integrating multiple learning paradigms into an ensemble architecture.

3.3 Inspiration for the Proposed Model

The proposed model draws inspiration from several prior studies, integrating their strengths to address the challenges of fall detection.

Hybrid Learning: Ma *et al.* [79] introduced a hybrid framework combining supervised and unsupervised components. Their system employs a thermal camera for privacy-preserving face localization, masking facial regions in RGB streams captured by an RGB camera. Spatiotemporal features are extracted using a 3D Convolutional Neural Network, while an AE detects falls through reconstruction errors. Although this approach effectively integrates both paradigms, it does not leverage their collaboration for unified decision-making, leaving room for further optimization.

Decision Fusion: The concept of decision fusion, as explored in sensor-based systems [15], inspires the integration of outputs from supervised and unsupervised models. Unlike Wang *et al.* [15], who combined outputs from different sensor modalities, the proposed model applies decision fusion within a meta-model to unify predictions from the AE and 3D CNN models.

Stacking Ensembles: Previous studies [24, 129] demonstrated the effectiveness of stacking architectures with multiple branches. The proposed model builds upon these approaches, streamlining the design into a two-branch structure that combines outputs from the AE and 3D CNN. This

simplified architecture balances performance and computational efficiency, making it suitable for real-world deployment.

Meta-Learners: Shukralia *et al.* [128] highlighted the superior performance of supervised meta-learners compared to individual models. This evidence supports the inclusion of a meta-model in the proposed framework to enhance classification accuracy and robustness, aligning with the goals of this research.

By synthesizing insights from these studies, the proposed stacking ensemble model advances fall detection by integrating supervised and unsupervised learning in a cohesive architecture. It aims to bridge the gap between high accuracy and practical real-world deployment, addressing the limitations of existing approaches while prioritizing robustness and scalability.

3.4 Methodology

Figure 3.1 provides a high-level representation of the proposed human fall detection ensemble model. The ensemble consists of three main components:

(i) **Supervised 3D CNN Model**: This model processes video data to detect falls based on previously observed fall patterns. It is trained on the fall and non-fall data split and outputs predictions on whether a fall is detected in the input frames.

(ii) **Unsupervised Autoencoder (AE) Model**: This model focuses on detecting anomalies in Activities of Daily Living (ADL) data. By training only on non-fall ADL activities, it identifies frames with reconstruction errors as potential anomalies, which may indicate falls.

(iii) **Meta-Model (MLP)**: The outputs from the 3D CNN and AE are combined and passed to a meta-model, a Multi-Layer Perceptron (MLP). This meta-model integrates the supervised and unsupervised predictions to classify frames as either a fall or not. This step mitigates the biases or limitations of individual models by leveraging their complementary strengths.

The figure also separates the **Training Procedure** from the **Real-Time Inference Procedure**. During training, the models are optimized on the respective splits of the dataset, while during real-

time inference, the meta-model uses the outputs of the AE and 3D CNN models to make the final classification. The colored blocks in the figure distinguish different phases: blue for preprocessing, red for intermediate training steps, green for the meta-model training, and tan for the real-time inference steps.

Specific information about data preprocessing and data generators is available in the supplemental repository: GitHub Repository.

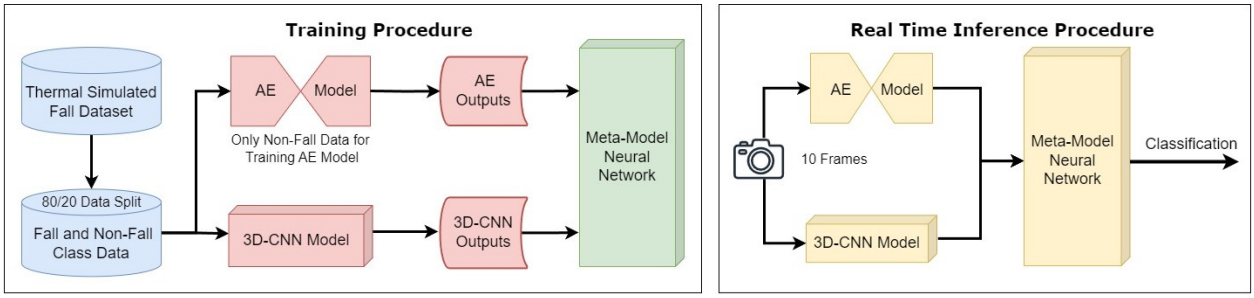


Figure 3.1: A high-level representation of the ensemble model.
Note: The Training Procedure (blue) involves dataset preprocessing, training individual models (red), and training the meta-model (green). The Real-Time Inference Procedure (tan) uses the trained AE and 3D CNN models to classify input frames as a fall or not. This structure balances supervised fall detection with unsupervised anomaly detection, leveraging the strengths of both approaches.

3.4.1 Model Architectures

3D CNN Model

Table 3.1 provides the layer-wise architectural detail of the 3D CNN. It receives a sequence of ten frames of size 256×256 and it is trained on both fall and non-fall video samples from the benchmark dataset summarized in Table 3.2. To address class imbalance, this work employs a sliding window-based data augmentation technique. By creating new samples that differ by one frame in the temporal dimension, this approach effectively balances the dataset. A 50:50 split between fall and non-fall events is used, with non-falls further divided into four sub-classes: an empty room, a person sitting, walking, and lying down. The supervised learning sub-network leverages both fall and non-fall samples to identify distinguishing patterns between classes. However, this approach requires a large volume of data due to its complexity. This limitation is mitigated through the

Table 3.1: Architectural detail of the 3D CNN model

Layer ID	Layer Type	Output Dimension
Input	Input Layer	(16, 10, 256, 256, 1)
L1	Conv3D	(16, 8, 254, 254, 32)
L2	MaxPooling3D	(16, 8, 127, 127, 32)
L3	Dropout	(16, 8, 127, 127, 32)
L4	Conv3D	(16, 6, 125, 125, 64)
L5	MaxPooling3D	(16, 6, 62, 62, 64)
L6	Dropout	(16, 6, 62, 62, 64)
L7	Conv3D	(16, 4, 60, 60, 128)
L8	MaxPooling3D	(16, 4, 30, 30, 128)
L9	Dropout	(16, 4, 30, 30, 128)
L10	Flatten	(16, 460800)
L11	Dense	(16, 64)
L12	Dropout	(16, 64)
Output	Dense	(16, 1)

Total # of trainable parameters: 29,768,897; Activation function: L1, L4, L7: ReLU; Output: sigmoid
Kernel size: (3,3,3) for Conv3D operations, (1,2,2) for MaxPooling3D layers; Padding: Valid padding is always used; Dropout rate: L3, L6, L9: 0.25; L12: 0.5; Learning rate: 0.001; Optimizer: Adam; # of Epochs: 75; Batch size: 16; Loss Function: Binary cross-entropy / log loss

Table 3.2: Summary of Thermal Simulated Fall Dataset [2]

Video Type	# of Videos	Total # of Frames	% of Total Frames
Falls	35	36,391	62.2
Non-Falls	9	22,116	37.8
Total	44	58,507	100

integration of unsupervised learning in the proposed solution. Figure 3.2 illustrates the 3D CNN model's training progress over 75 epochs, where it achieved optimal accuracy and minimized loss.

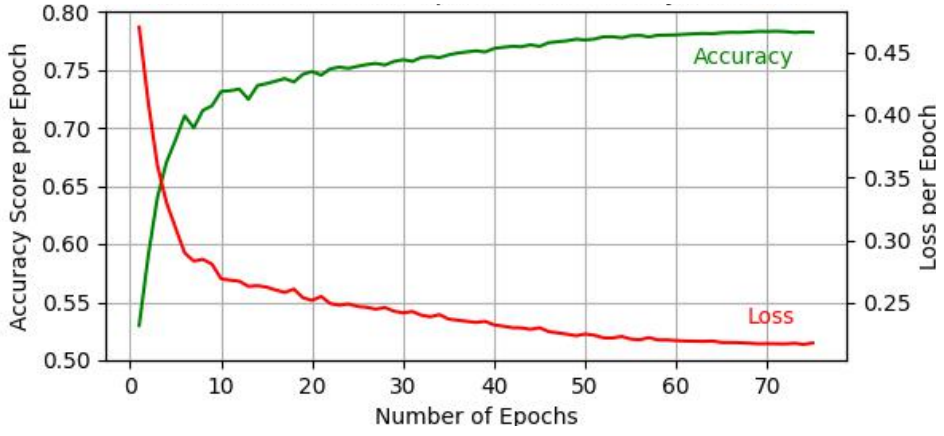


Figure 3.2: Training progress of the 3D CNN model for 75 epochs.

Table 3.3: Architectural detail of the AE model

Layer ID	Layer Type	Output Dimension
Input	Input Layer	(16, 10, 256, 256, 1)
Encoder L1	Conv2D	(16, 10, 256, 256, 8)
Encoder L2	MaxPooling2D	(16, 10, 128, 128, 8)
Encoder L3	Conv2D	(16, 10, 128, 128, 16)
Encoder L4	MaxPooling2D	(16, 10, 64, 64, 16)
Encoder L5	Conv2D	(16, 10, 64, 64, 32)
Encoder L6	MaxPooling2D	(16, 10, 32, 32, 32)
Decoder L1	Conv2D	(16, 10, 32, 32, 32)
Decoder L2	UpSampling2D	(16, 10, 64, 64, 32)
Decoder L3	Conv2D	(16, 10, 64, 64, 16)
Decoder L4	Upsampling2D	(16, 10, 128, 128, 16)
Decoder L5	Conv2D	(16, 10, 128, 128, 8)
Decoder L6	Upsampling2D	(16, 10, 256, 256, 8)
Output	Conv2D	(16, 10, 256, 256, 1)

All non-input layers are TimeDistributed layers to capture the temporal dependencies; Total # of trainable parameters: 12,785; Activation function: Encoder L1, Encoder L3, Encoder L5, Decoder L1, Decoder L3, Decoder L5: ReLU; Output: sigmoid; Kernel size: (2,2) for MaxPooling2D operations, (3,3) for Conv2D layers; Padding: Same padding is always used; Learning rate: 0.001; Optimizer: Adam; # of Epochs: 60; Batch size: 16; Loss Function: MSE

Autoencoder Model

Table 3.3 summarizes the architecture of the AE developed in this work. The AE processes the same input data as the 3D CNN model and operates as an unsupervised anomaly detector. It is designed symmetrically, with encoder and decoder modules leveraging convolutional operations to share spatial parameters. The AE is trained exclusively on ADL samples, representing non-fall events, to minimize reconstruction error on normal sequences. During inference, both fall and non-fall videos are provided to the model. The reconstruction error is calculated as the average squared difference between the pixel intensity values of the input frames and the AE-reconstructed frames. If this error exceeds the threshold established during training on non-fall events, the sequence is classified as an anomaly, indicating a potential fall. This approach enables the AE to detect deviations from normal patterns without relying on labeled fall data. However, it may flag certain non-fall actions as falls if those actions were not well-represented in the training data. To address this limitation, the AE's outputs are integrated with the supervised 3D CNN results in a co-learning framework, allowing the system to refine its predictions and improve overall accuracy.

Table 3.4: Architectural detail of the meta-model

Layer ID	Layer Type	Output Dimension
Input	Input Layer	(16, 2)
L1	Dense	(16, 128)
L2	Dropout	(16, 128)
L3	Dense	(16, 64)
L4	Dropout	(16, 64)
L5	Dense	(16, 32)
L6	Dropout	(16, 32)
L7	Dense	(16, 16)
L8	Dense	(16, 1)

Total number of trainable parameters: 11,265; Activation function: L1, L3, L5, L7: ReLU; L8: sigmoid; Dropout is always 0.2; Learning rate: 0.001; Optimizer: Adam; Number of Epochs: 25; Batch size: 16; Loss Function: Binary cross-entropy / log loss

The Meta-Model

The meta model is the final classifier built using an multi layer neural network as described in Table 3.4. It receives the intermediate class probability scores yielded by the earlier-mentioned, previously trained sub-networks and generates a refined classification output identifying fall and non-fall events in the input videos. Training the models separately circumvents any potential error propagation from the meta-model that could affect the weights of the 3D CNN and the AE. The neural network meta-model has 2 inputs, which are the outputs of the AE model and the 3D CNN model. This model is a dense feedforward network. The output layer has a sigmoid activation function. If the output of this activation function is above 0.5, there is a fall detected. Otherwise, the output of this activation function is below 0.5, there is no fall detected. Figure 3.3 displays the relationship between accuracy, loss, and the number of epochs during the training of the meta-model. The training is conducted over 25 epochs, as beyond this, the model overfits since the model accuracy on the validation data starts to decline while the loss begins to increase.

3.4.2 Network Justification

In order to determine the optimal architecture for each model, the same methodological approach for each of the three sub-networks was adopted, utilizing RandomSearchCV to whittle down potential hyperparameters, viz. the number of layers, their parameters, and activation functions among others. The initial values selected for RandomSearchCV drew inspiration from the current state-

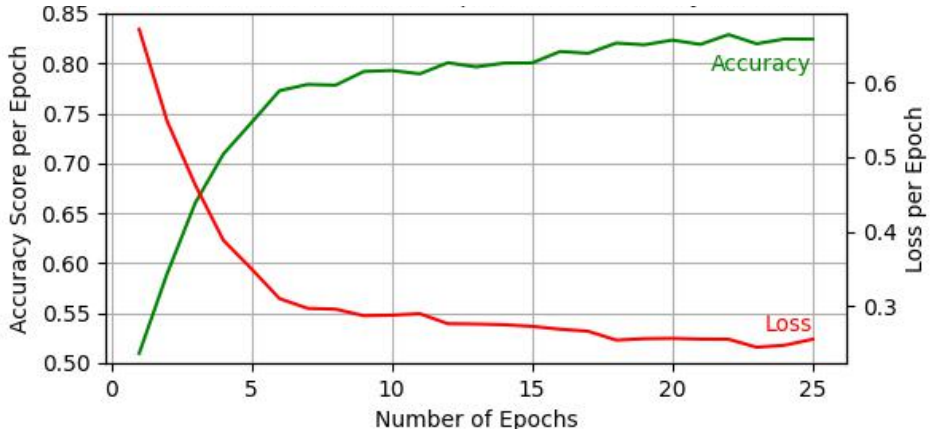


Figure 3.3: Training progress of the meta-model for 25 epochs.

Table 3.5: Distribution of fall videos in TSF Dataset between body position and fall direction

		Right	Left	Towards	Away
Total Data	Kneeling	4	2	0	0
	Sitting	1	2	5	1
	Standing	0	17	3	1
Testing Data	Kneeling	2	0	0	0
	Sitting	0	1	1	0
	Standing	0	2	1	0

of-the-art solutions. After pinpointing the most promising hyperparameter values using RandomSearchCV, GridSearch was subsequently applied to the condensed list, thereby identifying the optimal architecture.

3.5 Experimental Analysis

3.5.1 Environment

The proposed model is developed using Python version 3.10.11 and its open-source native libraries along with DL libraries, such as Keras with a TensorFlow backend. The model development, training, and testing are carried out on a system with an AMD Ryzen 7 5825U with Radeon Graphics 2.00 GHz processor and 16GB of RAM (13.9GB usable) connected to Google Colab. The training GPU is an NVIDIA Tesla K80 with 2496 CUDA cores and 12GB of VRAM.

3.5.2 Dataset

The publicly available "Thermal Simulated Fall" [2] (TSF) benchmark dataset (cf. Table 3.2) is used to validate the proposed human fall detection model. It contains successive .jpeg images constituting a thermal imaging video. Each image is 480 by 640 pixels in grayscale format. The frame distribution for fall and non-fall events is 62.2% and 37.8%, respectively. This indicates there is a class imbalance in the samples. It should be noted that data augmentation is not applied other than upsampling the data through the sliding window approach as mentioned earlier. Table 3.5 shows the distribution of fall samples based on the actions found inside each video. They are sorted based on the starting position of the human in each fall (vertical axis), and the position that they fall relative to the sensor (horizontal axis). This information is displayed for both the entire fall data samples and for the testing data.

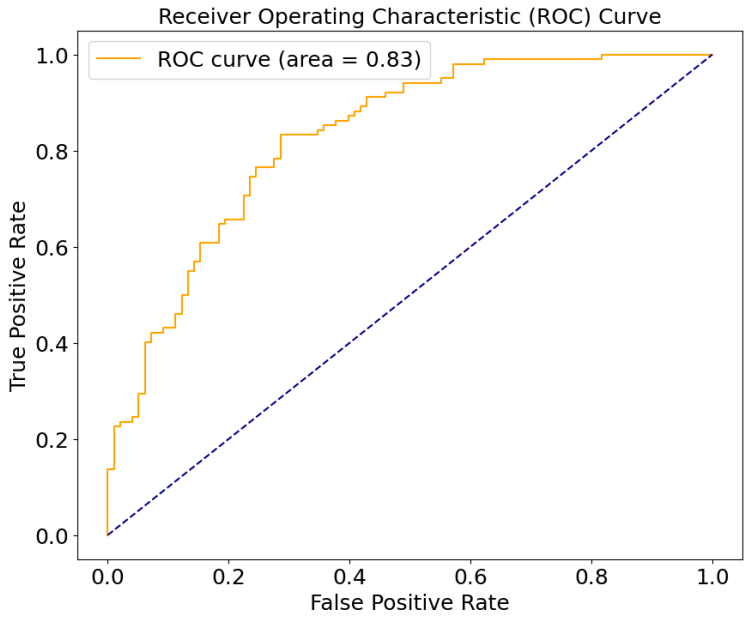


Figure 3.4: ROC-AUC of the meta-model that combines the outputs of the 3D CNN and AE.
Note: The AUC value of 0.83 indicates the meta-model is performing better than the individual 3D CNN and AE.

3.5.3 Quantitative Analysis

Table 3.6 compares the performance of the proposed solution with a recent baseline model introduced in [116] that is considered a defining piece of work switching the focus from supervised

Table 3.6: Comparative analysis of various models trained on the TSF Dataset [2]
Note: GFLOPS - giga floating-point operations per second, The mean (μ) and standard deviation (σ) indicate the anomaly score of the reconstruction error across frames

Model	ROC-AUC %	% Improvement	GFLOPS	Year
DAE [116]	64	$\downarrow$ 14.67	-	2018
CAE Deconv. [116]	75	Baseline	-	2018
ConvLSTM-AE (μ) [116]	76	$\uparrow$ 1.33	-	2018
ConvLSTM-AE (σ) [116]	83	$\uparrow$ 10.67	-	2018
CLSTMAE [118]	83	$\uparrow$ 10.67	-	2020
SRAE [118]	97	$\uparrow$ 29.33	-	2020
DSTCAE-C3D (μ) [76]	93	$\uparrow$ 24.00	-	2020
DSTCAE-C3D (σ) [76]	97	$\uparrow$ 29.33	-	2020
Adversarial learning (μ) [75]	95	$\uparrow$ 26.67	-	2021
Adversarial learning (σ) [75]	95	$\uparrow$ 26.67	-	2021
Fusion-Diff-ROI-3DCAE (μ) [70]	93	$\uparrow$ 24.00	-	2021
Fusion-Diff-ROI-3DCAE (σ) [70]	93	$\uparrow$ 24.00	-	2021
3D CNN (this work)	79	$\uparrow$ 5.33	17.7	2023
AE (this work)	74	$\downarrow$ 1.33	4.03	2023
3D CNN-AE (this work)	83	$\uparrow$ 10.67	21.76	2023

to unsupervised learning for human fall detection. The experimental study reveals promising performances of the proposed approach with an ROC-AUC score of 83%. The integrated model surmounts the individual models by 4% and 9% when compared to the 3D CNN and AE, respectively. The AE model performs slightly worse yet remains competitive with existing AE solutions, performing only 1% worse than the baseline CAE model in [116]. The 3D CNN model outperforms the baseline by 5.33% in terms of percentage improvement and doesn't have any direct comparisons in the literature with this dataset. The integrated model surpasses the baseline by 10.67% improvement and performs better than the individual AE model and 3D CNN model. This confirms the hypothesis that performance can be improved by feeding a stacking ensemble of different models into a meta-model classifier. The ROC-AUC curve in Figure 3.4 indicates that the model can attain a high TPR, albeit at the cost of an elevated FPR.

The GFLOPS (giga floating-point operations per second) computed are estimated during the inference time and not during the training time. The total number of floating-point operations required to perform a forward pass through the model is used. This is achieved by converting the input model into a TensorFlow graph and then using the TensorFlow profiler to count the total number of floating-point operations in the graph. Therefore, these estimates do not take

Figure 3.5: Randomly selected test samples showing the classification of fall and non-fall frames. Note: Two sets of 10 consecutive frames are shown to illustrate the model's classifications. The model correctly identified (A) starting at the 500th frame of the 7th fall video as a fall, and (B) starting at the 512th frame of the 5th non-fall video as a non-fall.

into account additional operations like backward propagation and weight updates, which are not required during inference procedures.

3.5.4 Qualitative Analysis

To visually inspect the performance of the proposed model, two randomly selected sets of ten consecutive distinct images from the test set are used and can be visualized in Figure 3.5. In each of these test cases the model correctly classified the samples. This reinforces the framework's appropriateness for privacy-protected human fall detection. Upon manual examination, the most common misclassified videos are the fall samples that contain someone falling to the right from their knees, and the samples that have someone falling to the left from sitting. Revisiting the sample distribution given in Table 3.5, the data is clearly skewed towards falls that begin from a standing position and fall towards the left. The ratio between testing data and overall data is quite high for the instances where the person was kneeling and fell to the right, and when they were sitting and fell to the left. This explains the common misclassifications, as there is likely not enough data in the training set for these types of falls to be represented sufficiently. Falls from standing were almost always classified correctly, following similar logic as 60% of the fall videos included a fall from standing. This analysis suggests that misclassification is primarily due to the data distribution rather than inherent flaws in the 3D CNN, AE, or meta-model. While techniques such as data augmentation and adding a preprocessing step to randomize training and testing samples in

each iteration can help alleviate this issue, the root cause lies in the small, imbalanced dataset. A more robust and balanced dataset will ultimately be needed to fully resolve these challenges.

3.6 Chapter Summary

This chapter demonstrates the effectiveness of a stacking ensemble framework combining a 3D CNN and an AE for privacy-preserving fall detection. The experimental results validate the hypothesis that integrating supervised and unsupervised models through a meta-model classifier can improve performance over individual models. The ensemble achieved an ROC-AUC score of 0.83, surpassing the individual 3D CNN and AE models by 4% and 9%, respectively, and outperforming baseline models from the literature.

Qualitative analysis further highlights the robustness of the framework, correctly classifying the majority of fall and non-fall samples. However, observed misclassifications are primarily attributed to the imbalanced data distribution, where certain fall types, such as those from kneeling or sitting positions, are underrepresented in the training set. Addressing this limitation through data augmentation or rebalancing could further enhance system performance and generalizability. Although the proposed system shows promising results, there is room for improvement. More sophisticated architectures could replace the current 3D CNN and AE to capitalize on advancements in deep learning. This, combined with the comprehensive thermal dataset proposed in the next phase of this research, has the potential to further improve fall detection performance, particularly for real-world deployments.

This phase of the research addresses the initial hypothesis that integrating supervised and unsupervised learning improves performance under constrained data conditions. However, it also underscores the critical need for a robust, diverse, and comprehensive dataset, as the lack of sufficient representation in the training data remains a bottleneck. The next phase of this thesis introduces the TF-66 dataset, designed to address these gaps, providing the field with a benchmark that supports both robust model evaluation and generalizability to real-world scenarios. By advancing both algo-

rithmic design and dataset development, this work aims to bridge the gap between research-centric fall detection systems and practical, deployable solutions capable of saving lives.

Chapter 4

Thermal Fall 66: Creation and Applications of a Novel Dataset

4.1 Overview

The Thermal Fall 66 (TF-66) dataset introduced in this chapter provides a comprehensive and diverse collection of fall scenarios recorded across multiple indoor environments with a wide range of participants. TF-66 addresses a significant gap in fall detection research by enabling researchers to improve the reliability and generalizability of fall detection systems in real-world environments. As the first publicly available indoor, occlusion-free, and privacy-preserving thermal dataset designed specifically for fall detection, TF-66 captures high-resolution (140×60) thermal images that ensure privacy is maintained and allow effective detection across various lighting conditions, including low or no light. Ceiling-mounted sensors provide a consistent overhead perspective across nine environments, mitigating occlusion issues common in wall-mounted systems and enhancing model generalizability. The dataset features 66 participants and diverse room configurations, incorporating varied room heights and furniture arrangements, along with extensive recordings of activities of daily living to improve fall vs. non-fall differentiation.

Unlike existing datasets, which often limit FDS effectiveness to the specific environments in which they were recorded, TF-66 is designed for broad generalizability across various real-world

settings. This generalizability addresses a core limitation in current FDS research, as models trained on previous datasets tend to perform well only in controlled or familiar environments [60]. By filling this gap, TF-66 not only enables reliable performance in diverse, unseen locations but also supports practical FDS deployment in real-world environments. Interestingly, the use of a more comprehensive dataset like TF-66 may initially lead to a drop in model performance compared to results on simpler, curated datasets. As Li *et al.* [115] observed, models achieving near-perfect sensitivity on datasets like URFD and MCFD saw their performance fall to 60–70% sensitivity on the more challenging High-Quality Fall Simulation Dataset. This discrepancy underscores the critical importance of using datasets that reflect real-world variability, revealing the inadequacy of over-optimized models designed for narrow, controlled conditions. TF-66's realistic scenarios are therefore a strength, ensuring that models trained on this dataset are better equipped to handle the complexities of real-world deployment and do not give the false impression that fall detection is a solved problem.

To support widespread deployment, TF-66 utilizes the Calumino CTS-EVK thermal sensor, offering an optimal balance of high resolution and affordability. This cost-effective sensor choice, validated through market analysis, ensures practical fall detection systems without the need for expensive multimodal setups. TF-66 comprises 562 fall videos (57,694 frames) and 250 non-fall videos (90,921 frames), recorded at 4 frames per second (fps) with a rigorous data methodology. Notably, TF-66 includes the highest number of fall samples among thermal fall detection datasets, providing DL models with a more representative training set to improve real-world installation performance [41, 60]. A customizable data generator accompanies TF-66, allowing researchers to adjust parameters such as video sequence length and dataset subsets, facilitating tailored model training for supervised and unsupervised learning approaches. Baseline experiments using a basic 3D CNN model on both the full dataset and individual subsets establish a performance benchmark, guiding future research and development.

The following sections outline the methodology behind TF-66, detailing data acquisition, sensor setup, participant diversity, and environmental configuration.

4.2 Data Acquisition Methodology

To ensure that the fall simulations in TF-66 accurately reflect real-world fall distributions, a detailed data acquisition methodology was developed for contextually rich recordings. The following subsection explains how falls were categorized, simulated, and recorded to mirror the characteristics of falls commonly experienced by elderly individuals.

4.2.1 Evidence-Based Fall Selection

A key challenge in fall detection research is capturing the wide variety of fall types within a single dataset [39]. While it is impractical to represent every possible fall orientation and condition, the TF-66 dataset adopts an evidence-based approach, similar to the UP-Fall dataset [55]. The falls in TF-66 are selected based on the types and frequencies documented in real-world studies. This method aligns with established trends in fall detection research, which categorize falls by direction relative to the individual's initial position [15, 130].

4.2.2 Fall Action Distribution

The fall simulations in TF-66 were carefully designed to reflect real-world scenarios, informed by a comprehensive study of 125 community-dwelling women aged 65 and older, who reported 158 fall events over one year [1]. Certain pre-fall actions from the study, such as "walking on uneven surfaces," "engaging in sports," "riding vehicles (e.g., bicycles)," and "stepping down stairs," were excluded due to limitations in the recording environments. After removing these actions, the remaining pre-fall events formed an adjusted distribution suitable for indoor simulations that can be easily matched. TF-66's fall action distribution closely replicates this modified real-world distribution, with all pre-fall actions within 2% of their real-world counterparts and most showing less than 0.5% variance. Figure 4.1 compares the real-world distribution to that of TF-66.

This accurate distribution matching was achieved using 12 structured "fall templates," developed in collaboration with a physiotherapist to guide recording technicians. Each participant performed falls according to a rotating template, with each template containing 12 distinct fall scenar-

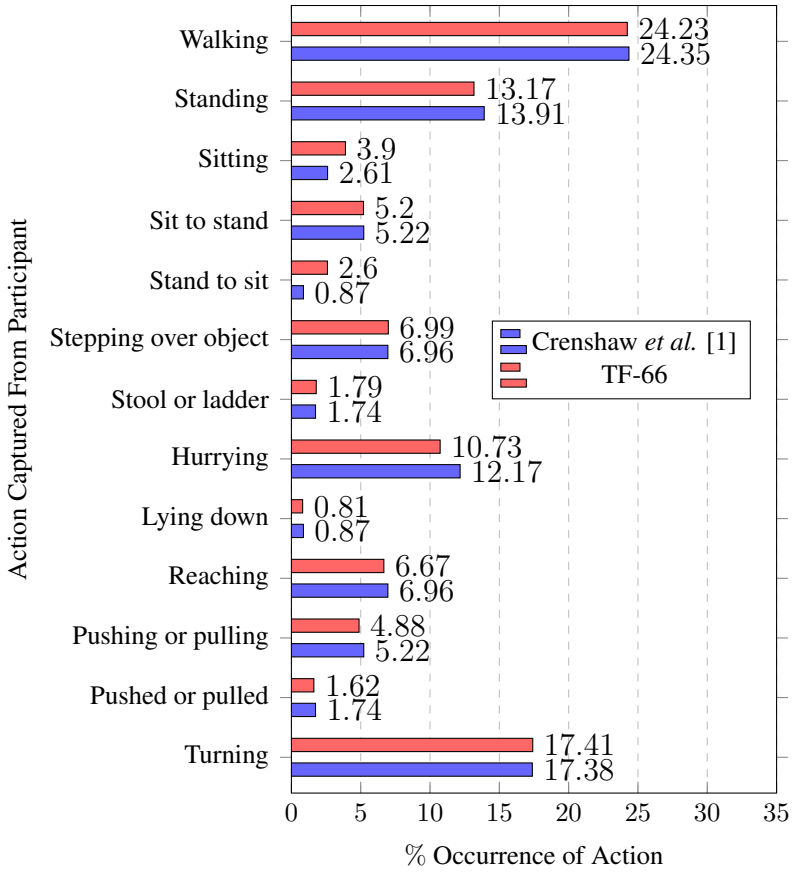


Figure 4.1: Actions that the participants were instructed to perform and their distribution in Crenshaw *et al.* [1] and TF-66.

ios. This structured approach ensured consistent adherence to the literature-based fall distribution across participants, accurately replicating real-world fall dynamics. In addition to pre-fall actions, the referenced study also provided statistics on fall directions and points of impact, which were incorporated into the TF-66 dataset design to further align with real-world fall dynamics. An example of a fall template used during recordings is provided in Table 4.1.

4.2.3 Direction of Falls and Point of Impact Distribution

TF-66 mirrors the fall direction distribution found in existing literature but has a slightly higher proportion of backward and forward falls, with fewer lateral falls (cf. Fig. 4.2a). These discrepancies will be addressed in future dataset versions to better align with real-world fall distributions. The dataset also measured points of impact during falls, showing a skewed distribution compared to

Table 4.1: Example of a fall template used to instruct actors during dataset creation

Note: This template was provided to recording technicians and physiotherapists to guide actors during dataset creation. The "Action," "Direction of Fall," and "Fall Onto" values were adjusted to reflect real-world fall distributions, as outlined in [1]

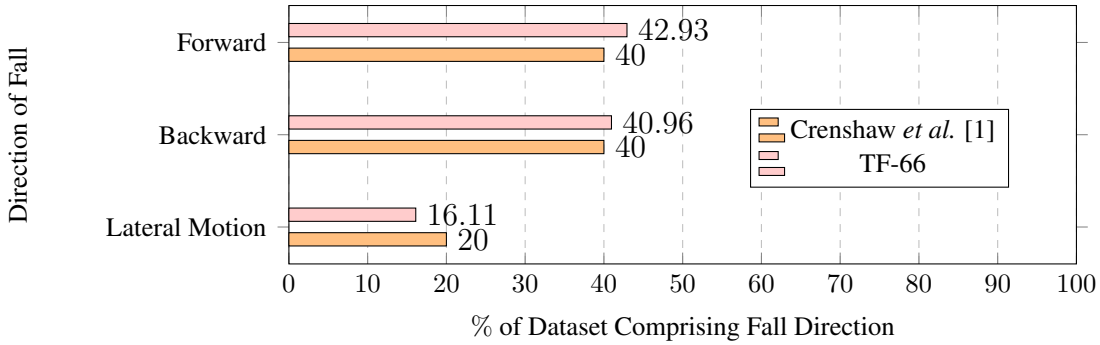
Fall Template	Action	Direction of Fall	Fall Onto
1	Walk	Forward	Furniture
2	Walk	Forward	Wall
3	Walk	Forward	Floor
4	Stepping Over Object	Forward	Floor
5	Reaching Out	Forward	Floor
6	Standing, Collapse	Backward	Floor
7	Standing, Collapse	Backward	Floor
8	Careless Hurrying	Backward	Floor
9	Changing Direction	Sideways	Floor
10	Changing Direction	Backwards	Floor
11	Standing Up	Sideways	Floor
12	Sitting Down	Sideways	Floor

real-world data (cf. Fig. 4.2b), with more instances of individuals landing on the floor or furniture and fewer involving railings. The imbalance stems from the exclusion of stair-related falls, which will also be addressed in future iterations of the dataset. Real-world alignment was achieved by incorporating fall direction and impact points into the rotating templates. These parameters were varied across templates, ensuring that the pre-fall actions were not repetitive for every fall direction and point of impact, thus enhancing the dataset's robustness and generalizability.

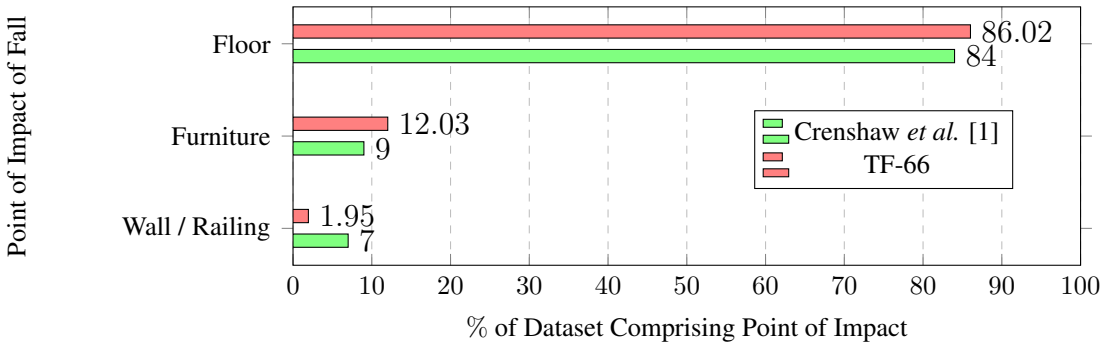
Grounding the selection of falls recoreded for the TF-66 dataset in real-world distributions, considering pre-fall action, fall direction, and point of impact, enables TF-66 to more accurately replicate the patterns and dynamics of actual falls. This provides models with a more realistic samples, improving their ability to accurately detect falls while minimizing false positives.

4.2.4 Simulated Falls Considerations

Although the ideal dataset for fall detection would consist solely of real falls, the rarity of fall events makes this approach currently infeasible. Simulated falls, while not fully representative of real-world falls, provide a practical alternative for constructing an initial dataset. These simulations enable the collection of a diverse range of fall events in a controlled and scalable manner. Natural falls are exceedingly rare. For instance, Stone *et al.* reported falls on just 0.3% of monitored days



(a) Directions of falls in the Crenshaw *et al.* [1] experimental study and in TF-66.



(b) Point of impact during falls in the Crenshaw *et al.* [1] experimental study and in TF-66.

Figure 4.2: TF-66 Dataset vs Crenshaw *et al.* [1] distribution.

[131], while Debard *et al.* observed falls on 1.6% of days [132]. Extrapolating these trends, monitoring 10 volunteers continuously for 5,000 days would yield only 500 natural falls, insufficient for robust classification [133]. Another study over 300 days with 15 seniors was conducted, with only 4 falls being captured, further illustrating the challenge of gathering natural fall data [18]. A longitudinal study found a fall rate of 1.3 falls per person per year among 125 community-dwelling women aged 65 and older [1]. Retaining elderly volunteers in studies is also difficult due to health issues where only one of three participants completed a study due to a death of one participant and cognitive decline of the other [18].

A significant challenge in fall detection is the class imbalance between fall and non-fall events. Falls last between 0.8 and 2.4 s [20, 123], while non-fall events dominate daily activities, creating an imbalance of about 12.13 million non-fall events for every fall [134]. Defining a "fall class" is complicated by the variability among fall events [39, 133]. Despite these limitations, simulated falls are widely accepted as suitable for fall detection research [18, 20, 55]. Studies have found

acceleration patterns in simulated falls of middle-aged adults similar to real falls in seniors, validating their use [18]. However, simulated falls must reflect seniors' experiences; they typically fall more slowly than younger individuals [9, 15, 20, 21]. To address this, a physiotherapist attended some of the recordings of the TF-66 dataset to guide participants in simulating falls, inspired by the expert input used in the creation of the eHomeSeniors dataset [20]. While this foresight is important, it is also vital for a comprehensive benchmark dataset to have some samples of seniors falling, to validate that the other simulated falls accurately depict falls that are similar to senior falls [9]. TF-66 meets this requirement with seven senior citizens participating in the data collection initiative. Although simulated falls are essential for developing fall detection systems, the long-term objective is to shift from training on simulated data to real-world fall data as detection technologies gain broader adoption.

4.2.5 Non-Fall Sample Inclusion Criteria

In the absence of specific studies outlining the real world occurrences of non-fall actions, the non-fall actions recorded for TF-66 were selected based on common daily activities performed by seniors. During some recordings, participants were instructed to perform tasks such as walking, sitting, and cleaning. To enhance the realism, the participants were also encouraged to engage in their typical actions without falling. To address the challenge of prolonged prone positioning, which heuristic-based fall detection often misclassifies as a fall, TF-66 includes non-fall sequences where participants remain motionless for at least 15 s. This helps distinguish genuine falls from extended inactivity. By aligning fall selection with real-world cases—considering pre-fall actions, fall direction, and impact points—TF-66 provides a more realistic foundation for DL models, improving fall detection accuracy and reducing false positives.

4.2.6 The Recording Environments

Samples were recorded in nine environments, as outlined in Table 4.2. A key strength of the TF-66 dataset is its diverse range of settings that closely mimic real-world scenarios, featuring

Table 4.2: The nine environments considered for fall event recordings in the TF-66 dataset
Note: Rm - Room, PCP - Participant, Hgt - Height

Rm ID	PCP ID	# PCPs	Rm Hgt.	# Falls	# NonFalls
1	1-14	14	8 Feet	90	45
2	15-16, 25-27	5	9 Feet	32	17
3	17-19	3	9 Feet	18	8
4	20-22, 28-37	13	9 Feet	139	47
5	23-24	2	9 Feet	13	7
6	38-44	7	10 Feet	82	30
7	45-46, 50-54, 61-62	9	8 Feet	67	28
8	58-60, 63-66	7	8 Feet	64	41
9	47-49, 55-57	6	8 Feet	57	27
Total		66		562	250

Table 4.3: The approximate size of the effective coverage area of the CTS-EVK in each size of the room found within the TF-66 dataset
Note: FC - floor coverage, which is the distance on the ground that is effectively captured by the lens of the CTS-EVK

Room Height	FC Length	FC Width	Coverage Area
8 ft	23 ft	13 ft	229 ft
9 ft	24.5 ft	14.75 ft	361 ft
10 ft	26.25 ft	16.5 ft	433 ft

common furnishings like tables and chairs. The first six environments consist of open layouts with strategically arranged furniture to simulate typical living spaces. The seventh environment is a simulated hospital lab with a central hospital bed and curtains, resembling a real hospital setting. The eighth mimics a long-term care home living room with a couch, chair, and table, while the ninth replicates a long-term care apartment bedroom with a bed, nightstand, and walker. For example, Figure 4.3 depicts the room dimensions and furniture configuration for Room 7, where the thermal camera is fixed on an 8-foot ceiling. Table 4.3 details the specific dimensions the camera can capture at various heights, indicating that effective sensor coverage increases with room height, which is crucial in clinical settings.

4.3 The Data Acquisition Device

For the assembly of the TF-66 dataset, the Calumino Thermal Sensor Evaluation Kit v3.1 was employed, which captures raw thermal imagery at a resolution of 35×15 pixels, with an in-application

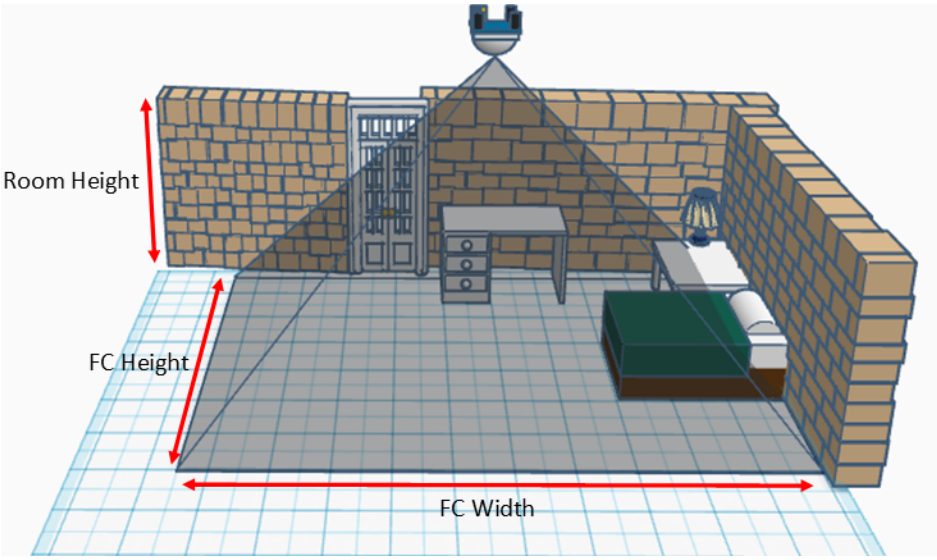
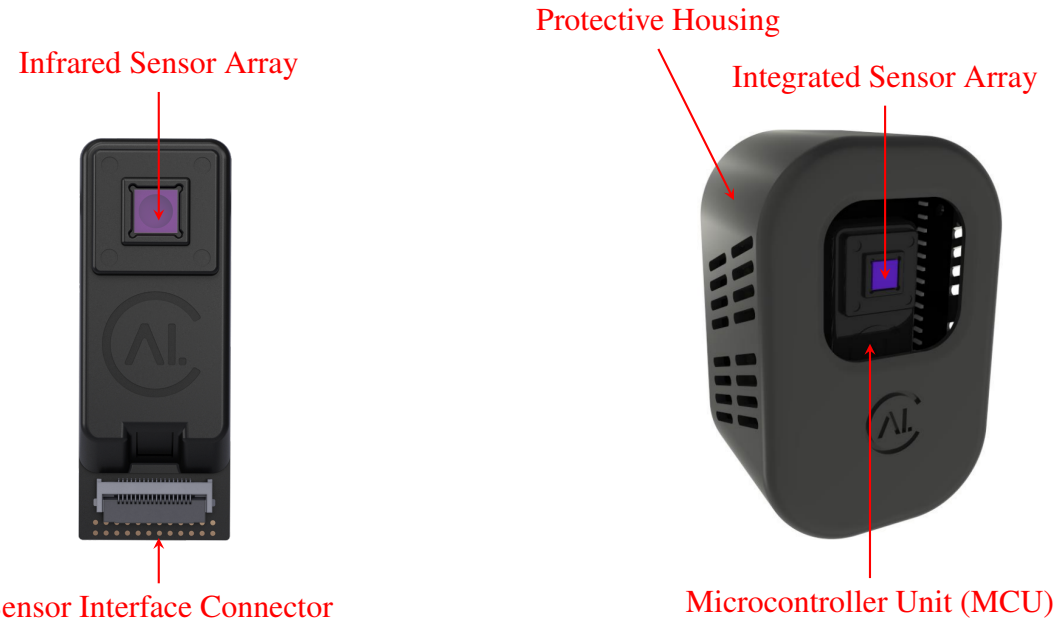


Figure 4.3: The schematic and environmental setup for Room 7 in the TF-66 dataset.
Note: This schematic illustrates the top-down thermal sensing approach used in the TF-66 dataset, with the shaded area representing the effective coverage (FC - floor coverage). The FC corresponds to the area captured by the CTS-EVK sensor, shown here for an eight-foot-high room.

option for image upsampling, thereby producing thermal images with a resolution of 140×60 pixels. An image of the zoomed in sensor array used is found in Figure 4.4a. The evaluation kit includes this sensor integrated on an ESP32 board and placed in a protective casing. This entire CTS-EVK can be seen in Figure 4.4b. The device is small and visually pleasing, mitigating potential product adoption issues due to bulkiness of the device. The CTS-EVK connects through USB or over Message Queuing Telemetry Transport (MQTT) to the Calumino desktop applications provided alongside the CTS-EVK, "CTS Player" which records raw data and "CTS Playback" which converts the recorded data to thermal videos. The CTS-EVK thermal sensor employed for this dataset includes a built-in human presence detector, which automatically marks the presence of individuals with a plus symbol in both real-time inference and recorded images. This helps mitigate background artifact issues, as the DL system can learn to associate the plus sign with human presence. This feature aligns with the ideal fall detection system requirement of incorporating a built-in human presence detector.



(a) The individual thermal sensor that captures an upsampled 140×60 pixels thermal image.

(b) The entire CTS-EVK unit that recorded the TF-66 dataset.

Figure 4.4: Key components of the CTS-EVK system used for thermal image capture in the TF-66 dataset.

4.3.1 Static Color Configuration for Consistency

To ensure uniformity across images and enable more reliable inferences in DL models, it is essential to establish a consistent configuration for thermal data visualization. The "CTS Playback" application includes a standardized color range feature, allowing users to input static "center", "max", and "min" values to control how thermal images are displayed. This static setting prevents dynamic rescaling and ensures that images remain comparable throughout the recording process. In TF-66, this feature was utilized for every sample, with the "center" value set to room temperature (30°C), "max" set to 30°C, and "min" set to 0°C. By maintaining these static values, TF-66 overcomes a limitation found in the MUVIM dataset [39], where dynamic color rescaling introduced variability across images, hindering model generalization.

4.3.2 Resolution Enhancement

As described in Figure 2.2, the CTS Player application offers built-in settings for automatic image upscaling to enhance detail. For example, Figure 2.2c clearly shows limbs and head shape, while lower resolutions (Figure 2.2b and Figure 2.2a) obscure details. TF-66 was recorded with $4\times$ upscaling, providing 140×60 pixel resolution. To be clear, this resolution enhancement is not a preprocessing step partaken after the data is collected, it is a configuration setting within the "CTS Player" application that is set prior to recording the data, that determines how the output images will appear.

4.4 Data Acquisition Process

The data acquisition process was carefully designed to balance authenticity and realism, creating robust and diverse samples while addressing the challenges of capturing realistic falls.

Participants provided written consent for their images to be used in research and commercial applications. Before recording, they registered online, agreeing to terms outlining study procedures and risks. Upon arrival, participants signed a consent form and completed a demographic survey capturing details such as age, gender, height, weight, medical conditions, as well as tracking the type of clothing and headgear they were wearing. This setup ensured effective categorization and analysis of data based on personal characteristics. Each 30-minute recording session was divided into two phases: non-fall activities and fall simulations. The first 10 minutes focused on typical activities of daily living, such as walking, sitting, lying down, and performing routine tasks like tying shoes or using a laptop. These provided a comprehensive dataset of non-fall behaviors. The remaining 20 minutes were dedicated to fall simulations, guided by 12 predefined "fall templates" developed based on protocols from prior studies. Participants rotated through different fall types and directions, with a physiotherapist present to ensure movements closely mimicked those of elderly individuals, who tend to fall more slowly and deliberately [15, 20, 21]. Each fall ended with 15 s of post-fall stillness to capture critical frames for classification purposes. Instructions detailed pre-fall movements, fall specifics, and post-fall behavior, ensuring consistency across sessions.

To ensure variety, room layouts, furniture placement, and crash mat positions were randomized for each session. Falls were recorded with start and end frames logged in an Excel sheet, accessible on GitHub. Re-enactments addressed technical issues, incomplete data, or corrupted recordings. Safety was prioritized with cushioned mats, optional assisted falls, and the freedom to withdraw without penalty. No injuries occurred, and participants were treated with care. The resulting dataset surpasses existing benchmarks with enhanced thermal sensor capabilities, built-in human detection, and greater diversity in environments and participants.

4.5 Data Curation

Each sample in the TF-66 dataset was recorded as an `.avi` video file using the CTS Playback desktop application, which allowed for the specification of frame numbers to ensure adequate context before and after each fall event. This context is crucial for reducing false positives by helping the model distinguish falls from actions such as bending over, based on the duration the participant remains on the ground. Each fall video includes a manually recorded timestamp in an accompanying spreadsheet, marking the start of the fall. Ideally, videos contain 40 frames preceding and 70 frames succeeding the fall, translating to approximately 10 s before and 17.5 s after the fall at a recording rate of 4 frames per second. However, some videos may have fewer frames due to recording issues or variations in participant performance. On average, fall videos last 25.31 s, slightly less than the expected 27.5 s. Any discrepancies in video lengths have been addressed, as described in the Data Generator section (cf. Section 4.10). Immediately after recording, the data was processed with playback settings adjusted to match the room temperature at the time of recording. Each sample was cropped, converted into an `.avi` file, and renamed following the naming convention illustrated in Figure 4.8, which is found in the Dataset Organization section (cf. Section 4.8). Non-fall/ADL samples underwent minimal preprocessing, limited to setting the "center" variable to room temperature. The complete data acquisition workflow, from participant registration to video upload, is depicted in Figure 4.5.

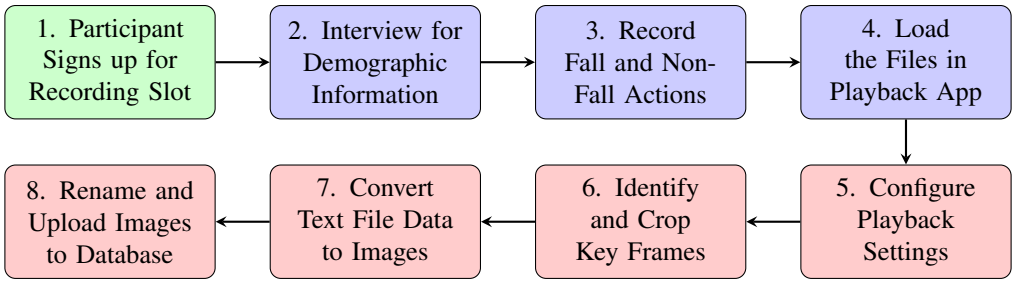
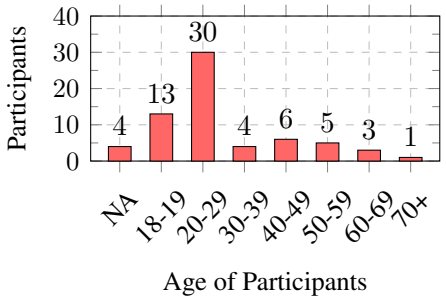
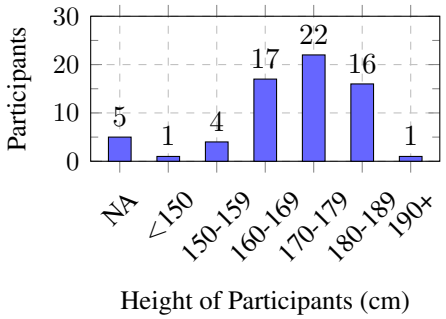


Figure 4.5: The data acquisition process.
Note: The green, blue, and red blocks occur before, during, and after the recording session, respectively.

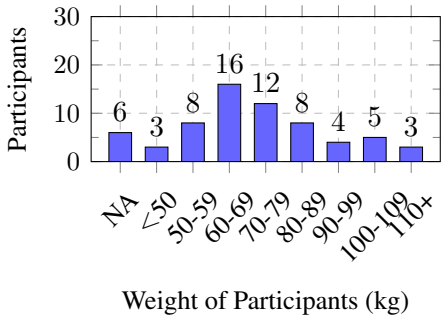
4.6 Demographic Overview of TF-66



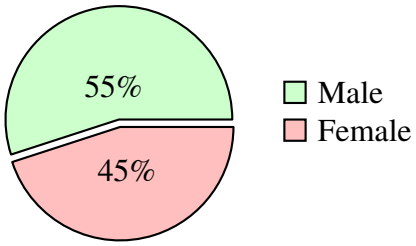
(a) The age distribution of participants.



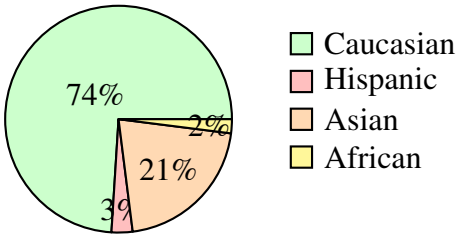
(b) The height distribution of participants.



(c) The weight distribution of participants.

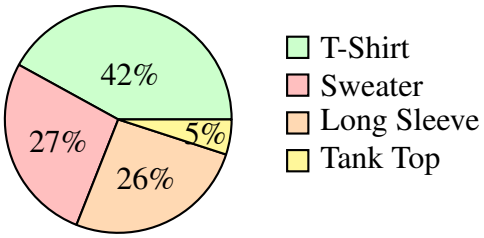


(d) The gender distribution of participants.

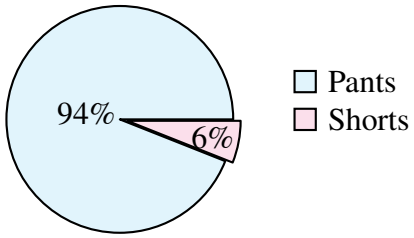


(e) The declared ethnicity of participants.

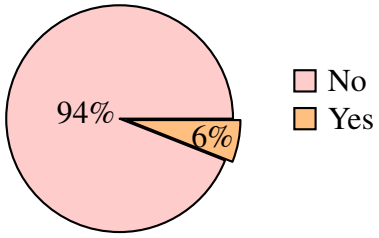
Figure 4.6: Participant demographic summary in TF-66.



(a) The upper body clothing worn by participants.



(b) The lower body clothing worn by participants.



(c) Participants wearing head coverings.

Figure 4.7: Types of clothing worn by participants in TF-66.

Demographic information for each participant was collected prior to the recording sessions, summarized in Figure 4.6. As shown in Figure 4.6a, most participants are under 30 years of age, with a mean age of 30.075 years. Figure 4.6b illustrates the height distribution, approximating a normal distribution. Figure 4.6e reveals a bias toward Caucasian participants, though this imbalance is not expected to significantly affect fall behaviors, aside from potential cultural influences on fall speed.

There was an almost equal distribution of male and female participants (36 male and 30 female) as demonstrated in Figure 4.6a, minimizing bias related to movement speed and body size. Self-declared weights of participants were also tracked, which follows a near-normal curve with some heavier outliers as shown in Figure 4.6c.

Figure 4.7 provides an overview of participant clothing during recordings, essential for assessing the impact on thermal imaging effectiveness. Figure 4.7a shows that 31 of the 66 recording participants had exposed arms during recording, while Figure 4.7b indicates that long pants predominated, with shorts worn in only 4 recording sessions. Figure 4.7c highlights head coverings, with 4 participants (numbers 4, 25, 39, and 58) wearing various headgear. Including participants with head coverings enhances the dataset's relevance for scenarios where such items are common,

given the head's significance as a thermal signature, especially when the sensor is mounted on the ceiling. Participants were also screened for health conditions or medications affecting their fall risk. Notably, one participant had Multiple Sclerosis and a history of falls, while another had Type 2 Diabetes and Hypertension. These conditions did not appear to influence fall behaviors during recordings. All health-related information remains anonymous to protect participant privacy.

4.7 The Specialized Subsets of TF-66

The TF-66 dataset includes specialized subsets designed to address specific fall detection research needs, enhancing both accuracy and generalizability. These subsets enable researchers to tailor algorithm training to the specific environments where systems will be deployed. As detailed in Table 4.4, the subsets are categorized by room height (e.g., eight-foot, nine-foot, and ten-foot ceilings), participant demographics (e.g., a subset focusing on seniors), and unique recording environments (e.g., hospital settings in Room 7). This level of customization helps mitigate model bias caused by environmental variations and supports the development of more precise and context-specific fall detection solutions. In ceiling-mounted thermal fall detection systems, the size and intensity of a person's thermal signature change based on their distance from the sensor. For instance, a person lying on the floor appears smaller than one on elevated surfaces, like a bed, as they are closer to the sensor and therefore appear larger. This variation becomes even more significant across rooms with different ceiling heights. In a 10-foot room, for example, a person lying on a couch might produce a thermal signature similar to the thermal signature produced if they were lying on the floor in an 8-foot room. Training with height-aligned data enhances model performance, making height-based subsets essential. The "senior" subset focuses on fall data from older adults, capturing unique characteristics like slower movements, while the "hospital" subset addresses specific scenarios, such as patient falls from beds. Future expansions of the TF-66 dataset will include a staircase subset to address this common fall risk area, along with further representation in underrepresented subsets like the ten-foot room category.

Table 4.4: TF-66 subset summary: the table summarizes participant groupings and the number of falls recorded across subsets defined by ceiling heights or specific participant categories (e.g., seniors, hospital settings).

Subset	Participant ID	# Participants	# Falls
8 Feet	1-14, 45-66	36	278
9 Feet	2-5	23	202
10 Feet	38-44	7	82
Senior	13, 38, 43-44, 53-54, 56	7	65
Hospital	45-46, 50-54, 61-62	9	67

4.8 Dataset Organization

The dataset structure of TF-66 is illustrated in Figure 4.8. The root contains "Train" and "Test" folders, organized to reflect an 80/20 split between training and testing data. This division was carefully crafted, not only ensuring an 80/20 split in terms of the number of videos but also maintaining this ratio for the total number of frames. For consistency and unbiased model comparisons, it is recommended that researchers use the pre-divided training and test data when conducting experiments. Within both "Train" and "Test" folders, there is a "Fall" folder and a "NonFall" folder, separating each type of recording. Within each of those folders, there are subfolders containing an individual fall or non-fall video. Each subfolder follows this naming convention: `X-Y-Z`, where `X` is the participant ID for the recording sample, `Y` is "NonFall" if the video in question is a NonFall, otherwise it is "Fall", and `Z` is the sequence number of `Y` samples for participant `X`.

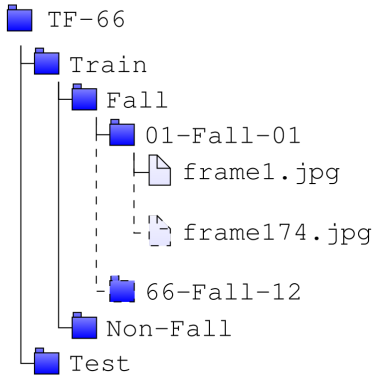


Figure 4.8: The directory organizational structure of the TF-66.

4.9 Qualitative Analysis of Dataset

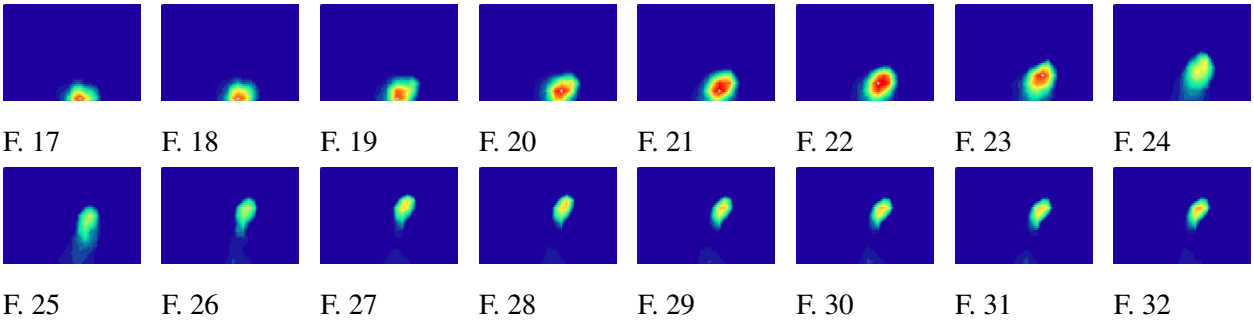


Figure 4.9: 16 consecutive frames from 38-Fall-01 starting at frame 17.
Note: The frames depict a person walking into the scene, falling onto their stomach, and remaining in a prone position.

Figure 4.9 demonstrates the benefits of using a ceiling-mounted thermal dataset for fall detection. The thermal imaging inherently anonymizes individuals' identities, which is a major privacy advantage compared to datasets that combine thermal and visual images, potentially revealing identities. The ceiling-mounted sensor also reduces occlusion problems often seen with wall-mounted sensors. Additionally, Figure 4.9 illustrates the clear changes in thermal signatures that indicate a fall. In frames 17 to 22, the heat signature is concentrated and strong, as the sensor is close to the individual's head, a main heat source while the person is walking. In frames 23 to 26, during the fall, the heat signature becomes spread out and weaker because the sensor captures the whole body from a further distance, effectively indicating a fall. The individual's continued presence lying down (in frames 27 to 32) with a continued weak heat signature in the following frames further confirms that a fall occurred, rather than just someone bending over. The human presence indicator, shown as a plus symbol on the images, was not annotated or added post-recording; instead, it is an optional feature automatically included in all images recorded using the "CTS Player" application. This indicator helps mitigate false alarms by distinguishing human heat signatures from non-human sources, which do not display this marker. Generally, falls can be confirmed if they meet the following criteria: (i) A dense, circular heat signature quickly moves to a more dim, elongated heat signature, (ii) This heat signature remains elongated and in a similar position for

subsequent frames, confirming the individual fell and is not simply bending over. Both of these criteria are met for the fall sample found in Figure 4.9, and all falls found in TF-66.

4.10 Use of Dataset

TF-66 is a publicly available dataset for non-commercial uses. Instructions on how to access the dataset have been provided in a GitHub–*TF-66*. It includes the data generator, `DataGenerator.ipynb`, the required dataset spreadsheet file, `Final Dataset.xlsx`, and a `ReadMe.txt` explaining key components of the data generator and caching techniques used.

4.10.1 Data Generators

To facilitate the use of the TF-66 dataset, customizable data generators have been developed that can be employed to easily produce batches of data from TF-66 for training and testing fall detection DL models. These generators streamline access to the dataset while providing researchers with flexible options for preparing input data for their models. The advantage of using them is that they ensure fair and unbiased model comparisons across different studies, allowing researchers to evaluate model performance on a consistent experimental setup that better reflects real-world conditions. These generators allow researchers to toggle dataset subsets and adjust video length with a single variable, supporting tailored experiments. Fall and non-fall video outputs are balanced, with each fall video containing the event and relevant contextual frames before and after. The length of generated clips doesn't need to match the full recording. The `num_frames` variable controls video length, and researchers can adjust this for their models.

As discussed in the Data Curation section, not all fall videos have exactly 40 frames before and 70 frames after the fall event. To manage these inconsistencies, an Excel file, "Final Dataset.xlsx," provides detailed information on each video, including discrepancies in frame counts. The data generator references this file when randomly selecting a video, using the corresponding `firstFallFrameOfVideo` value to calculate the start and end points for the generated video. It then creates two variables, `start_min` and `start_max`, representing the earliest

and latest possible start frames for the video to ensure that the fall event and the surrounding context are captured. A random frame within this range is selected as the starting point, and `num_frames-1` consecutive frames are loaded from that point onward. This approach guarantees that every fall video generated by the data generator contains the fall event along with relevant surrounding contextual information, making the dataset more robust and adaptable for machine learning research. It is imperative that all researchers utilize the same data generator with the manually annotated spreadsheet values for consistent evaluation of the dataset [81, 135]. Deviating from this standard by using a custom data generator could result in fall videos where the fall event is not actually found within a generated fall video, minimizing the effectiveness of the model. This inconsistency would hinder the ability to realistically compare models, as the inclusion of varying contextual frames before or after the fall event could influence the system's performance in unpredictable ways. Therefore, to ensure fair and transparent model comparisons, adherence to the standardized generator is crucial for all evaluations.

Sampling Interval

When passing data to models, some researchers have experimented with increasing the sampling interval to accelerate computation, thus not passing every consecutive frame to the model, but instead skipping frames at regular intervals. However, this approach has been shown to degrade model performance [61, 136]. The reason for this is likely due to the rapid nature of falls, where skipping frames can cause small yet critical details to be missed, making it essential to pass consecutive frames to models for accurate evaluation.

The literature presents varying recommendations on the optimal number of consecutive frames to use as input for fall detection models as a smaller number can increase false alarms, but a larger number can include more non-fall frames in the "fall" sample [76]. One study compared the use of 4 and 8 consecutive frames and found that using only 4 frames led to an increased number of false alarms [75]. The datasets used in this study were collected at frame rates between 15fps and 30fps, yet it was noted that the longest fall event lasted just 13 frames, meaning most falls occurred in less than one second. Another researcher explored different window lengths of 6, 8, and 10 frames,

concluding that 8 frames, equivalent to 1 second of video in their 8fps dataset, provided the best results [11]. On the other hand, the creators of the UP-Fall dataset recommended a much longer window length, including a 6-second safe period after the fall event as advised by [137].

This work adopts a balanced approach between the varying recommendations found in the literature. In this analysis, a window length of 10 frames is used, which corresponds to 2.5 seconds of real time, as TF-66 was recorded at 4fps. However, it is recommended that researchers experiment with different window sizes to determine the optimal length for maximizing model performance in their specific use case. The `num_frames` variable in the data generator code can be easily adjusted to modify the number of frames included in each sample, providing flexibility for model experimentation. The average fall video in the TF-66 dataset is 25.31 s long, giving researchers ample room to experiment with varying video lengths. Due to the extended video duration, researchers can also customize the amount of pre- and post-fall frames by adjusting the `start_min` and `start_max` calculations in the data generator code. This allows for further fine-tuning of the temporal context provided in each video clip, ensuring that models can be trained with the most contextually relevant data for fall detection.

Dataset Balancing

The TF-66 dataset contains 57,694 fall frames and 90,921 non-fall frames, but this imbalance is further amplified during data generation. The generator focuses on producing fall videos that include only the frames surrounding the fall event, effectively reducing the number of fall frames used for training. This class imbalance is critical, as it can hinder model training by skewing gradient calculations and leading to biased predictions [30]. For instance, an imbalanced dataset may result in a deceptively high ROC-AUC score by predominantly predicting the majority class (non-fall), giving a false impression of model performance. To address this issue, a dynamic balancing mechanism ensures equal representation of fall and non-fall videos during batch generation. For example, in a batch of 400 videos, 200 are randomly selected fall videos and 200 are non-fall videos. Once the quota for either class is reached, additional videos from that class are excluded from the batch. This ensures that no class exceeds 50% of the batch, thereby preventing bias. This

approach, inspired by Nunez-Marcos *et al.* [30], is particularly useful for mitigating imbalance by allowing both classes to be equally learned during training. Unlike prior methods that resampled non-fall data to match fall data, this solution is more practical when the fall frames are significantly fewer, as in TF-66. While standardized metrics such as the ROC-AUC curve can partially address the effects of class imbalance, this work also employs a comprehensive set of evaluation metrics to ensure a fair and nuanced assessment of model performance. An expanded version of this is discussed in Section 5.2.9.

Further Customization Options

The focus group results, previously shown in Figure 2.1c, highlighted a near-even split in user preferences: some prioritized high detection accuracy despite the risk of false alarms, while others preferred fewer false alarms, even if detection accuracy decreased. This feedback underscores the need for personalized fall detection settings tailored to individual preferences. To address this, TF-66 includes longer video segments, enabling researchers to adjust videos with the data generator based on user tolerance for false alarms. Future iterations will introduce three sensitivity levels to accommodate varying preferences, like, (i) **High sensitivity**: For users who tolerate frequent false alarms, ensuring quick recoveries are still detected as falls, (ii) **Medium sensitivity**: For users who prefer fewer false alarms, requiring the user to stay on the ground for several s before triggering an alert, and (iii) **Low sensitivity**: Minimizes false alarms, triggering an alert only if the user remains prone for an extended period.

The dataset currently has limited samples of "recovery fall" events, where someone falls, quickly gets up, and moves away. These would be classified differently based on the sensitivity value chosen by the senior, to match their false alarm and detection rate preference. This is especially important since existing fall detection systems and opinions in the literature seem to go back and forth on whether these "recovery fall" events should alert the associated caregiver when this action has been detected [52]. Future data collection will focus on capturing these events to ensure better representation. Overall, the data curation and generator parameters allow models to be fine-tuned according to individual user preferences and the specific environment where the

system is installed. This should not only improve system performance, but also improve system adoption and ultimately help more people through customizable fall detection solutions.

4.10.2 Data Caching

Bottleneck Identification and Analysis

Initially, a significant bottleneck was identified when generating new batches of data using the generators. Profiling revealed that the `load_img` method from Keras was responsible for this, taking 0.5 s to load each image on average. Given the training dataset contains 120,000 samples, this bottleneck could lead to a processing time of approximately 60,000 s (˜16.67 Hr) per epoch, solely for invoking the `load_img` method, assuming all samples are used.

Effective Solution through Caching

To address the issue, an attempt involved converting all images in the dataset into `NumPy` arrays and loading them with the `np.load` function. However, profiling revealed that this method unexpectedly increased the average batch generation time. The solution was to implement caching to reduce the overhead of reading images from disk by preloading images into memory. This approach minimized disk I/O operations during training and optimized execution, achieving a remarkable speedup of nearly $95\times$ compared to the baseline image-loading approach. It also reduced the overall storage size of the dataset, enhancing performance and efficiency. A detailed comparison of execution times, speedups, and dataset sizes for the original image loading, `NumPy` arrays, and cached images is provided in Appendix C (Table C.1). Once the cache is created, it can be reused for all subsequent dataset operations, eliminating the need for re-creation. The script and related helper files are available on the GitHub, with detailed instructions provided in Appendix C.

While caching effectively eliminates the bottleneck, the user must generate the cache locally on their machine. The cache is tied to the file path structure and directory where it was created and cannot be transferred between systems with different directory structures. For example, a cache generated on a system using Google Drive cannot be used on a different environment, such as the

Digital Research Alliance of Canada. In such cases, the cache must be regenerated within the target environment to align with its file path structure. This caching solution ensures faster, more efficient training times, while also accommodating the flexibility required for diverse research setups.

4.11 Model Experiments and Analysis

To set a benchmark for DL model performance on the TF-66 dataset, six versions of a 3D CNN were trained, either on the full dataset or individual subsets, to compare performance across subsets. The model was developed in Python 3.10.11 using Keras with a TensorFlow backend. Training and testing were conducted on the Cedar core of Compute Canada, utilizing an NVIDIA Tesla K80 GPU (2496 CUDA cores, 12GB VRAM).

4.11.1 Proposed Model

Figure 4.10 depicts the model and its layer details are summarized in Table 4.5. The training lasted up to 100 epochs, with early stopping applied after 10 epochs if no improvement in validation loss was observed. Input images were of size 256 $\times$ 256, and the dataset was mutually divided into an 80/20 train-test split. Binary cross-entropy (4.1) was used as the loss function, a standard choice

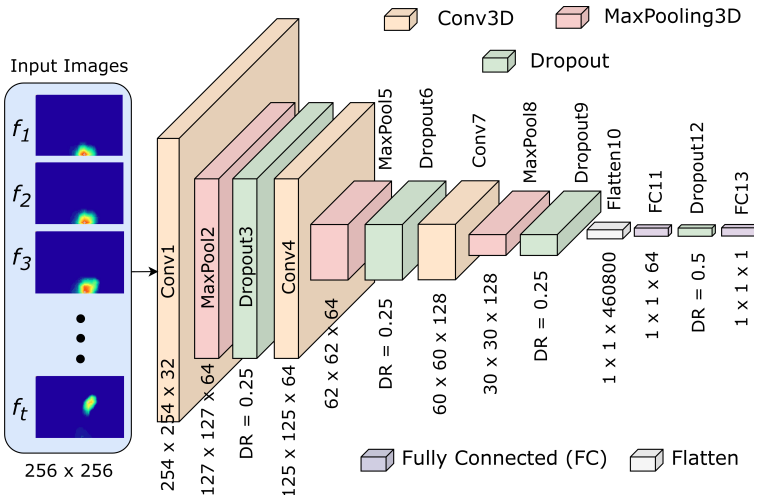


Figure 4.10: The architecture of the 3D CNN model employed on TF-66 and its subsets.

Table 4.5: Architectural Detail of the 3D CNN Model

Layer ID	Layer Type	Output Dimension
Input	Input Layer	(16, 10, 256, 256, 1)
L1	Conv3D	(16, 8, 254, 254, 32)
L2	MaxPooling3D	(16, 8, 127, 127, 32)
L3	Dropout	(16, 8, 127, 127, 32)
L4	Conv3D	(16, 6, 125, 125, 64)
L5	MaxPooling3D	(16, 6, 62, 62, 64)
L6	Dropout	(16, 6, 62, 62, 64)
L7	Conv3D	(16, 4, 60, 60, 128)
L8	MaxPooling3D	(16, 4, 30, 30, 128)
L9	Dropout	(16, 4, 30, 30, 128)
L10	Flatten	(16, 460800)
L11	Dense	(16, 64)
L12	Dropout	(16, 64)
Output	Dense	(16, 1)

Total number of trainable parameters: 29,768,897; Activation function: L1, L4, L7: LeakyReLU (alpha = 0.1); Output: sigmoid; Kernel size: (3,3,3) for Conv3D operations, (1,2,2) for MaxPooling3D layers
Padding: "Same" padding is always used; Dropout rate: L3, L6, L9: 0.25; L12: 0.5; Learning rate: 0.0001; Optimizer: Adam; Number of Epochs: 100 ; Batch size: 16; Loss Function: Binary cross-entropy / log loss

Table 4.6: The results of the proposed 3D CNN models on the TF-66 subsets

Subset	Epochs	ROC-AUC %	F1-Score %	MCC %	Accuracy %	Loss
TF-66	21	92.9	80.4	67.6	84.5	0.357
10ft	21	95.2	90.6	82.1	90.7	0.324
9ft	14	90.2	79.7	55.1	75.6	0.617
8ft	17	89.6	73.2	59.5	81.5	0.426
Hospital	16	98.5	83.4	79.7	93.7	0.140
Senior	18	99.4	91.7	90.1	97.3	0.073

for binary classification tasks.

$$\text{BCE} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(p_i) + (1-y_i)\log(1-p_i)\right] \tag{4.1}$$

4.11.2 Experimental Results

The proposed 3D CNN model was trained and evaluated six times: once on the entire dataset and once for each subset. For subset experiments, validation was performed exclusively on the corresponding subset's validation data. The results, summarized in Table 4.6, reveal several notable findings. Interestingly, subsets with fewer samples (e.g., 10-foot room, hospital environment, and

senior living environment) generally outperformed subsets with more data (e.g., 8-foot and 9-foot room height subsets). This contradicts the typical expectation that larger training datasets improve generalization. For example, the hospital subset achieved an impressive ROC-AUC value of 0.985 on the unseen validation set. This strong performance can be attributed to the subset's homogeneity: participants were often lying in bed, and falls predominantly involved rolling or collapsing in predictable patterns. In contrast, the 8-foot room subset, despite being part of the hospital environment, performed the worst, with a particularly low F1-Score. This may be due to the subset's higher diversity in fall scenarios, spanning various daily activities, which likely challenged the model's ability to generalize across these contexts. This suggests that dataset consistency can significantly impact model performance. The model trained on the entire dataset achieved a strong overall ROC-AUC value of 92.9%. However, anomalies were observed, such as an abnormally high loss on the 9-foot subset validation data. This may indicate potential bias or overfitting specific to this subset.

4.12 Chapter Summary

Existing fall detection datasets often suffer from limited generalizability, leaving researchers uncertain whether improvements observed on these datasets translate to advancements in real-world applications. These datasets frequently lack the robustness and diversity required for practical use, often relying on sensors that fail to capture the high-resolution data necessary for reliable fall detection. The TF-66 dataset addresses these limitations by utilizing the CTS-EVK sensor, which offers the highest resolution among economically viable thermal sensors suitable for large-scale deployment. TF-66 establishes a new benchmark in the field, surpassing existing datasets in participant diversity, environmental variety, and data quality, while maintaining privacy and minimizing occlusion-related issues. Moreover, it introduces tailored subsets to enable model optimization for specific deployment scenarios, enhancing the practical applicability of fall detection technologies. Accompanying the dataset is a comprehensive data generator that streamlines experimental workflows. This generator allows researchers to easily modify sequence lengths, adjust parameters, and select specific subsets, significantly simplifying the setup process. By providing a versatile

and comprehensive resource, TF-66 is poised to catalyze advancements in fall detection research globally, serving as a foundational benchmark for the field.

The following chapter will discuss the development of a more advanced and optimized fall detection model, evaluated on TF-66, its subsets, and the TSF dataset, further demonstrating the practical value of this dataset.

Chapter 5

The Advanced Fall Detection Model

5.1 Overview

This chapter builds upon the foundational work presented in previous chapters by systematically optimizing supervised models for thermal fall detection and addressing key limitations through innovative enhancements. While Chapter 3 explored hybrid approaches combining supervised and unsupervised methodologies, the development of the robust TF-66 dataset enabled this chapter to focus exclusively on supervised models. This work involved testing hundreds of model variations, each meticulously designed to assess the individual and combined effects of advanced techniques, such as batch normalization, various attention mechanisms (spatial, temporal, feature-based, self, and general), recurrent neural networks, Convolutional LSTMs (ConvLSTMs), Bidirectional Convolutional LSTMs (BiConvLSTMs), and optical flow integration.

Each component was independently optimized to identify its ideal configuration before combining them in diverse configurations to evaluate their collective impact on model performance. Two distinct data generation strategies—brute-force and balanced—were evaluated to examine their effects on training dynamics, generalization, and real-time feasibility. This exhaustive experimentation led to the identification of two optimal configurations: the ConvLSTM + Optical Flow model, which demonstrated superior performance on the TF-66 dataset, and the BiConvLSTM +

Attention Mechanisms model, which achieved state-of-the-art results with a ROC-AUC of 99.7% on the TSF dataset.

These findings underscore the value of systematic experimentation in advancing thermal fall detection. By establishing TF-66 as a benchmark dataset, this chapter provides valuable insights into the trade-offs between computational complexity, real-time feasibility, and model performance, offering a robust foundation for future research and practical deployment.

5.2 Model Training and Optimization Strategies

The initial training strategy for this chapter involved enhancing the individual components introduced in Chapter 3—namely the 3D CNN, autoencoder (AE), and a meta-model combining these architectures—using advanced optimization techniques. This approach was initially considered promising, particularly given the lack of a robust benchmark dataset for thermal fall detection. However, the development of the TF-66 dataset addressed many of these limitations, offering a significantly larger and more generalizable dataset for model evaluation. Despite its relatively small size compared to standard deep learning datasets, TF-66 provided sufficient diversity to support meaningful model optimizations and comparisons. The optimization process began with the 3D CNN model, which was selected due to its foundational role in thermal fall detection. Using the basic 3D CNN implementation from Chapter 3 as a baseline, hyperparameters were systematically adjusted to maximize performance. Each hyperparameter was evaluated independently, with all others held constant, to determine its unique contribution. This iterative approach allowed the identification of an optimal configuration that balanced performance and generalization. The robustness of the TF-66 dataset, combined with a brute-force data generation strategy, enhanced generalization further by repeatedly exposing the model to diverse fall scenarios, enabling effective feature learning. The thermal input modality and the consistent ceiling-mounted perspective provided additional benefits, reducing variability across samples and simplifying the generalization process with fewer examples. These improvements emphasized the effectiveness of supervised learning approaches when paired with high-quality datasets. Attempts to integrate the optimized

3D CNN model with autoencoders and meta-model ensembles were also explored. These hybrid approaches aimed to combine supervised and unsupervised techniques to improve performance further. However, their integration introduced significant trade-offs. Inference times for these combined models often exceeded the 250ms threshold critical for real-time fall detection applications. Furthermore, none of the joint approaches surpassed the performance of the independently optimized 3D CNN. As a result, the focus shifted entirely to supervised techniques, leveraging the robustness of the TF-66 dataset and advanced architectural enhancements.

This chapter continues by detailing the systematic optimization of the 3D CNN architecture, followed by advanced techniques such as the integration of ConvLSTM and attention mechanisms. These methods are evaluated through experiments on the TF-66 dataset, highlighting their contribution to achieving state-of-the-art performance in thermal fall detection.

5.2.1 Training Regularization and Efficiency

To prevent overfitting, early stopping with a patience of 10 epochs and minimizing the validation loss of the BCE was applied. Models converged to their optimal configurations within 4 to 59 epochs, with the best weights saved and restored for final evaluation. Mixed precision training was employed throughout to enhance computational efficiency for these larger models.

5.2.2 Hyperparameter Optimization

The hyperparameters with the most significant impact on performance included model depth, 3D CNN kernel size, padding, learning rate, batch size, activation functions, optimizer, and dropout rates. Key findings from this testing are summarized below:

- **Model Depth:** Increasing the depth of the 3D CNN by adding layers improved performance up to a point, but excessive depth led to vanishing gradient issues, especially given the relatively small size of the TF-66 dataset [63]. Shallower networks performed better, aligning with findings for similar datasets [81].

- **3D CNN Kernel Size:** A kernel size of $3 \times 3 \times 3$ offered the best balance between capturing fine details and identifying complex spatial-temporal patterns [136].
- **Padding:** Using "same" padding instead of "valid" padding ensured that input dimensions were preserved across layers. This adjustment significantly improved temporal feature extraction by maintaining the integrity of the temporal channel throughout the model, thereby enhancing overall performance.
- **Learning Rate:** A learning rate of 0.0001 outperformed higher rates (0.001 and 0.01), offering better stability and convergence.
- **Activation Functions:** Leaky ReLU with an alpha of 0.1 provided the best results, mitigating vanishing gradient issues and reducing training time compared to other functions such as ReLU and tanh [120].
- **Batch Size:** Optimal results were achieved with batch sizes of 16 and 32, while smaller sizes caused overfitting, and larger sizes failed to minimize loss effectively.
- **Dropout Rates:** A dropout rate of 0.25 after convolutional layers and 0.5 after fully connected layers effectively reduced overfitting.
- **Optimizers:** The ADAM optimizer consistently outperformed others, with RMSProp following closely. SGD showed limited utility for this application [120].

By combining these optimized hyperparameters, a refined baseline model was developed. This extended baseline, differing from the original 3D CNN in Chapter 3 by its activation functions and padding, was evaluated on the TF-66 dataset. Subsequent sections detail the independent evaluation of advanced techniques and their combined effects on model performance.

5.2.3 Addition of Batch Normalization

Batch Normalization (BN) is a widely utilized DL technique that stabilizes training by normalizing layer inputs to zero mean and unit variance within each mini-batch. BN mitigates internal

covariate shift, accelerates convergence, and incorporates learnable parameters to scale and shift normalized values, maintaining model flexibility. During inference, running averages of mean and variance are used to ensure consistent behavior [118]. In thermal fall detection with 3D CNNs, BN offers significant advantages. They have the ability to stabilize training for high-dimensional inputs, reducing the risk of vanishing or exploding gradients, and acting as a regularizer to prevent overfitting—an important consideration for small datasets. BN also can enable the use of higher learning rates, thereby accelerating training without degrading performance.

This work tested BN applied after each 3DConv layer, with variations that omitted BN layers or adjusted dropout rates. The configuration with BN after all 3DConv layers and mixed with a halving of all the dropout rate values achieved the best BN-specific results. However, this configuration underperformed compared to the baseline model, leading to the abandonment of BN-focused approaches.

5.2.4 Integrating Attention Mechanisms

Attention mechanisms were integrated into the 3D CNN model to enhance its ability to prioritize relevant spatial and temporal information [136]. These mechanisms improve the model's capacity to capture critical patterns by directing focus to specific features within the data, thereby addressing challenges posed by the dataset's uniform backgrounds and dynamic fall events. Spatial attention focuses on isolating the individual within each frame [61], while temporal attention models motion dynamics across frames, such as transitions from standing to falling. Together, these components enable the model to integrate global dependencies while preserving essential details.

The effectiveness of these attention mechanisms was validated through systematic testing. Feature-based attention emphasizes motion and posture indicative of falls, while self-attention captures complex relationships between spatial and temporal dimensions, improving both interpretability and accuracy [34]. However, the use of self-attention in isolation revealed a tendency to overemphasize specific features, leading to single-class predictions. To address this, residual connections were introduced to improve gradient flow and preserve input features, while dropout was applied to mitigate feature dominance. Despite these efforts, the residual connections reduced

performance below the baseline or introduced computational challenges, leading to the discontinuation of this enhanced self-attention configuration. Instead, a simplified self-attention mechanism with residual connections was successfully implemented and achieved competitive results, as detailed in Section 5.3.3.

Below, the mathematical formulations and placement of attention mechanisms are described, highlighting their contributions to model performance.

Spatial Attention

The spatial attention layer, $\mathcal{L}_{\text{attn}}^{\text{spatial}}$, generates attention weights by reducing the input feature map $\mathbf{X} \in \mathbb{R}^{B\times T\times H\times W\times C}$ along the temporal and channel dimensions. This reduction produces a 2D spatial feature map $\mathbf{F}_{\text{spatial}} \in \mathbb{R}^{B\times H\times W}$. A sigmoid activation function is applied to compute attention weights, which are then broadcasted and multiplied element-wise with the input:

$$\mathcal{L}_{\text{attn}}^{s}(\mathbf{X}) = \sigma\left(\text{Mean}_{\text{temp, channel}}(\mathbf{X})\right) \odot \mathbf{X}, \tag{5.1}$$

where σ denotes the sigmoid activation function, and $\odot$ represents element-wise multiplication. This mechanism effectively directs the model's focus to relevant spatial regions in each frame.

Temporal Attention

The temporal attention layer, $\mathcal{L}_{\text{attn}}^{\text{temporal}}$, captures motion dynamics across sequences by reducing the input feature map $\mathbf{X}$ along spatial dimensions, producing a temporal feature map $\mathbf{F}_{\text{temporal}} \in \mathbb{R}^{B\times T\times C}$. A sigmoid function is applied to compute attention weights, which are broadcasted and applied to the input feature map:

$$\mathcal{L}_{\text{attn}}^{t}(\mathbf{X}) = \sigma\left(\text{Mean}_{\text{height, width}}(\mathbf{X})\right) \odot \mathbf{X}, \tag{5.2}$$

where the attention weights normalize temporal contributions, enhancing the model's ability to track motion.

Feature-Based Attention

Feature-based attention, $\mathcal{L}_{\text{attn}}^{\text{feature}}$, emphasizes informative channels by aggregating spatial and temporal features into a channel descriptor $\mathbf{z} \in \mathbb{R}^{B \times C}$. Fully connected layers process $\mathbf{z}$ for dimensionality reduction and restoration:

$$\mathcal{L}_{\text{attn}}^{f}(\mathbf{X}) = \mathbf{X} \odot \sigma \left(W_{\text{restore}} \cdot \text{ReLU} \left(W_{\text{reduce}} \cdot \mathbf{z} + b_{\text{reduce}} \right) + b_{\text{restore}} \right), \tag{5.3}$$

where trainable parameters W and b refine the attention weights. A reduction ratio of 32 was found optimal, balancing computational cost and performance.

Self-Attention

The self-attention mechanism, $\mathcal{L}_{\text{attn}}^{\text{self}}$, refines global dependencies by modeling relationships across all dimensions. The input tensor $\mathbf{X}$ is reshaped, multi-head attention is applied, and the attended features are reshaped back:

$$\mathcal{L}_{\text{attn}}^{\text{self}}(\mathbf{X}) = \text{MultiHeadAttention}(\text{Reshape}(\mathbf{X}), \text{Reshape}(\mathbf{X})). \tag{5.4}$$

Optimal hyperparameters were identified as two attention heads and a key dimension of 128 to balance model complexity and performance.

Attention Placement and Impact

Inspired by [138], attention mechanisms were strategically placed: spatial attention was applied after the 1st `Conv3D` block, temporal attention after the 2nd block, and feature-based and self-attention after the 3rd block. This separation of spatial and temporal mechanisms, as demonstrated in video classification tasks, significantly enhanced performance by allowing each mechanism to focus on distinct aspects of the input data. However, self-attention required hyperparameter tuning to address memory constraints.

Through these combined efforts, the integration of attention mechanisms improved the model's ability to prioritize critical features across spatial, temporal, and channel dimensions, resulting in a robust fall detection system tailored to the dataset's characteristics.

5.2.5 Exploiting RNNs for Temporal Feature Learning

Given their sequential modeling capabilities, RNN-based mechanisms were investigated for extracting long-term temporal features following the 3D CNN's convolutional layers [61, 115, 136]. Although falls occur over short time scales, integrating LSTM-based approaches after flattening the 3D CNN output could help extract more temporal features from the data. Among RNN subtypes, LSTMs, GRUs, and Bi-LSTMs were considered, with GRUs eventually being excluded due to consistently lower performance [19].

To address spatial limitations, ConvLSTMs were tested. By employing convolutional operations in input-to-hidden and hidden-to-hidden connections, ConvLSTMs effectively propagate spatiotemporal features and are better suited for video-based inputs [61]. A novel Bi-ConvLSTM configuration was also explored, combining bidirectional processing with spatiotemporal learning. Experiments showed that the Bi-ConvLSTM module outperformed the baseline without additional attention mechanisms. When combined with a general attention mechanism, all RNN variants surpassed the baseline, with ConvLSTM performing best, followed by Bi-ConvLSTM, Bi-LSTM, and LSTM, respectively.

5.2.6 Incorporating Optical Flow

Optical flow was incorporated to provide additional motion information between consecutive frames, enabling the model to better focus on dynamic changes while ignoring static background features. Optical flow calculates the difference between two frames, highlighting areas of motion. For example, during a fall, optical flow captures normal motion, rapid motion as the person falls, and then no motion as they remain prone. This approach enriches the model's understanding of motion dynamics and has been shown to improve performance in prior studies [70]. Thermal-based op-

tical flow offers advantages over RGB-based methods, as it is unaffected by lighting changes and non-heat-emitting objects, ensuring more reliable motion detection. However, challenges remain, such as distinguishing human motion from that of animals and accounting for residual heat left behind by stationary individuals, which can create misleading optical flow patterns [8].

This work employed the Farneback method from the `cv2` library to generate optical flow images, systematically optimizing its parameters for the TF-66 dataset. Details of these tests are provided in Appendix D. Optical flow extraction was performed offline to reduce computational burden. Integrating optical flow as an additional input channel improved performance beyond the baseline model, validating its utility for fall detection.

5.2.7 Combination of Advanced Techniques

To further optimize the enhanced baseline model, advanced techniques, including RNN subtypes (e.g., ConvLSTMs and Bi-LSTMs), attention mechanisms (spatial, temporal, feature, and self-attention), and optical flow, were systematically combined. Each technique was incrementally added or removed in a controlled manner, ensuring that every possible combination was tested. This systematic ablation study aimed to identify the most effective configuration. From these experiments, four configurations emerged as the best-performing models:

- **ConvLSTM with General Attention and Optical Flow:** This configuration added a ConvLSTM layer after the final Conv3D layer of the 3D CNN, followed by a general attention mechanism that spanned globally across the input data. Optical flow was integrated as an additional input channel alongside thermal imaging.

- **BiConvLSTM with Layer-Specific Attention Mechanisms and Optical Flow:** Similar to the first configuration, but with a BiConvLSTM replacing the ConvLSTM. Additionally, spatial, temporal, and feature attention mechanisms were applied after the first, second, and third Conv3D layers, respectively.

- **ConvLSTM with Layer-Specific Attention Mechanisms and Optical Flow:** This mirrored the second configuration but replaced the BiConvLSTM with a standard ConvLSTM.

- **Self-Attention with the Baseline Model:** The baseline model was augmented with a self-attention mechanism placed after the 3rd Conv3D layer and before the fully connected layer.

The results of these configurations, detailed in Table 5.2 in Section 5.3.3, demonstrate the combined impact of these advanced techniques.

5.2.8 Cross-Dataset Evaluation

To evaluate the generalizability of the proposed models, the most successful configurations from the TF-66 dataset were tested on the TSF dataset, a widely recognized benchmark for thermal fall detection. This cross-dataset evaluation provided a direct comparison with state-of-the-art models, offering valuable insights into the robustness and transferability of the proposed methodologies. The results are summarized in Table 5.1 in Section 5.3.3. The TSF dataset [2] served as a complementary evaluation platform, validating the methodology and highlighting the adaptability of the models to different datasets. Widely used in benchmarking studies, TSF offers a reliable basis for comparing fall detection models across the field. To maintain consistency with prior studies, the ROC-AUC metric was used as the primary performance measure, ensuring standardized and meaningful comparisons.

The data generator designed for the TF-66 dataset was also employed for the TSF dataset, with one key modification: the sampling rate was adjusted to select every third frame instead of consecutive frames. This adjustment addressed the TSF dataset's higher frame rate of 12 fps, ensuring that 10-frame samples captured the entire fall event along with contextually relevant pre- and post-fall data. Testing revealed that consecutive frames at the TSF's native frame rate failed to provide adequate temporal representation of fall events.

5.2.9 Comparing Data Generator Approaches

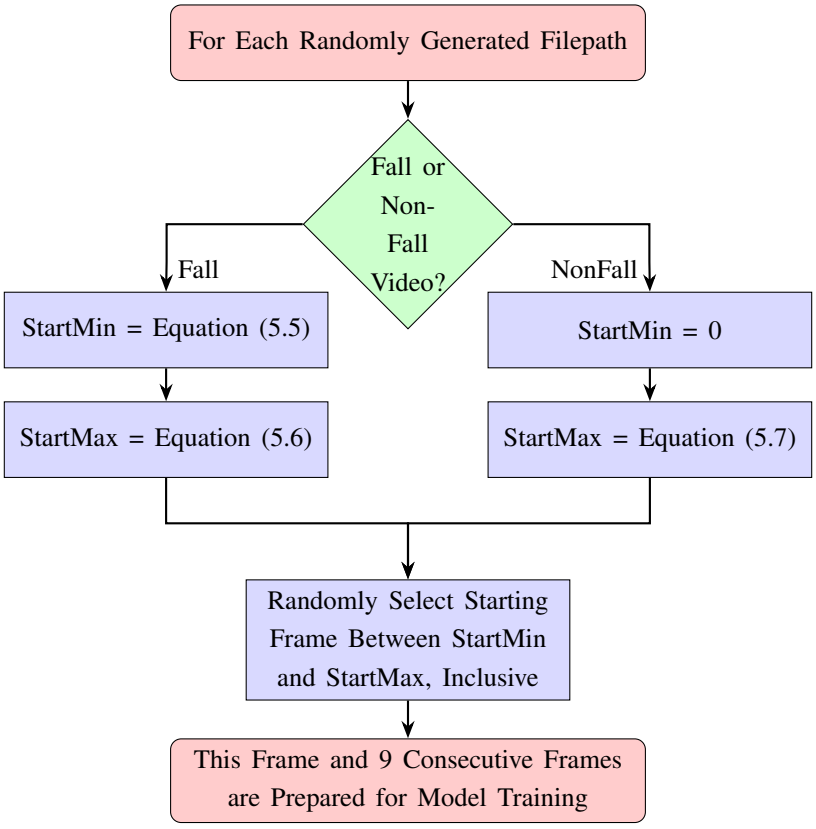
With the optimal model architectures identified, the focus shifted to evaluating the data generator's configuration to assess its impact on performance. Two distinct approaches to data generation were explored, differing in how samples were generated during each epoch.

Brute-Force Approach In the Brute-Force Approach, an epoch consisted of as many samples as there were filepaths to training images within the training dataset, with the generator selecting a random filepath for each sample. To construct a sample, the generator extracted 10 consecutive frames starting from a frame randomly selected within a defined range. For fall videos, this range was determined using metadata from the `Final Dataset.xlsx` file, ensuring that each sample included the fall event and its surrounding context. This process is detailed in the flowchart in Figure 5.1, with corresponding equations presented in (5.5)–(5.7).

While this method ensured comprehensive coverage of fall events, it resulted in significant overlap among generated samples. For example, in the TF-66 dataset, each fall video (110 frames long) produced 110 samples, but only 6 unique 10-frame sequences were possible, leading to an average of 18.33 repetitions per unique sequence per epoch. This issue was even more pronounced in the TSF dataset, where fall videos (1040 frames long) yielded 65 repetitions per sequence. Despite concerns about overfitting, testing showed that models trained using this approach performed well on unseen validation data. Results from this method are labeled as the "Brute-Force Approach" in the quantitative analysis (Section 5.3.3).

Balanced Approach To address the potential redundancy of the Brute-Force Approach, a Balanced Approach was implemented. Here, an epoch terminated once all unique samples for the minority class had been generated. The total number of unique samples was calculated as the product of the possible starting frames per video (Equations (5.8) and (5.9)) and the number of videos in the class. For the TF-66 dataset, this ensured six unique samples per fall video, while the TSF dataset allowed for 16 unique samples per fall video due to its higher frame rate. Figure 5.2 is provided for a visual representation of how these constraints are set and managed.

To mitigate class imbalance, batches were constructed to contain an equal number of fall and non-fall samples. A shuffled vector determined the class of each sample generated, ensuring both randomness and balance. While some samples might be repeated or missed in a single epoch, the law of large numbers ensured balanced representation across multiple epochs. Results from testing on both types of data generator approaches can be found in the quantitative analysis (Section 5.3.3).



(a) Flowchart illustrating the logic of dynamically generating the random frames passed to the DL models, ensuring relevant details are included within each video.

$$start\_min = \max\left(0, 1stFallFrame - numFrames - 1\right) \tag{5.5}$$

$$start\_max = \min\left(1stFallFrame - \left\lfloor \frac{numFrames}{2} \right\rfloor - 1, \left\lfloor \frac{totalFrames}{2} \right\rfloor - numFrames\right) \tag{5.6}$$

$$start\_max = totalFrames - numFrames \tag{5.7}$$

(b) Equations defining the calculation logic. "1stFallFrame" is the manually validated frame number where the fall event begins, "numFrames" is the number of frames being passed to the DL model, which is always 10 in this work, "totalFrames" is the total number of frames within the randomly selected video.

Figure 5.1: Flowchart and equations illustrating the logic for frame selection in model training. Figure 5.1a shows the flow of operations, while Figure 5.1b details the corresponding equations.

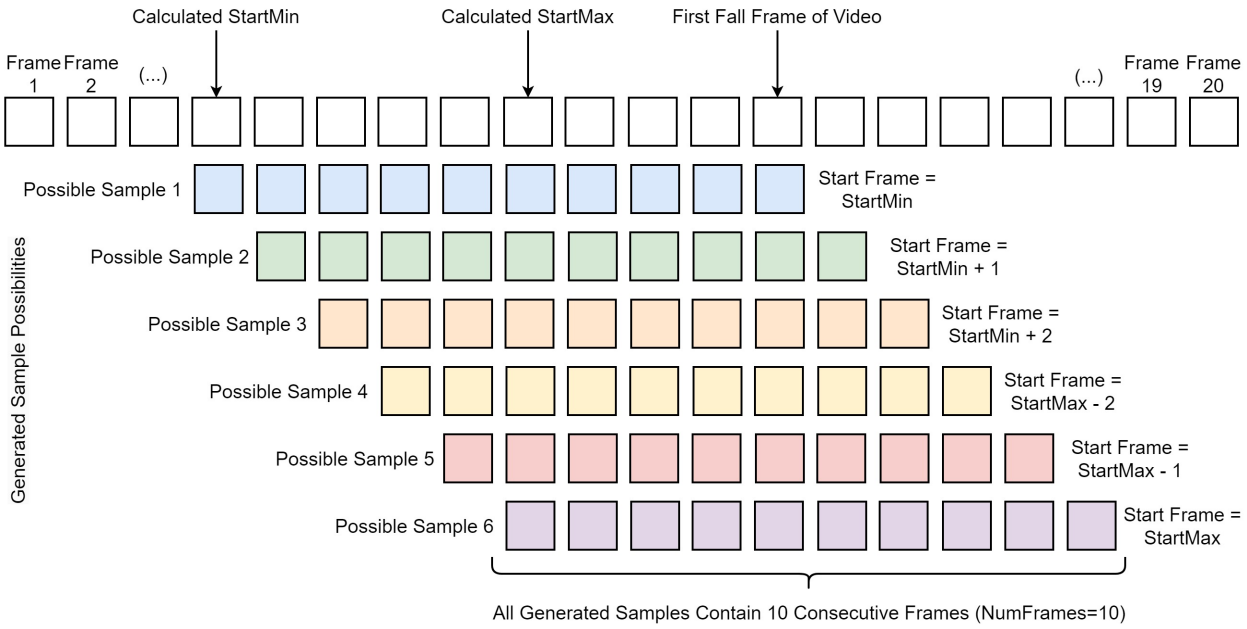


Figure 5.2: Illustration of the data generator used to produce sequential fall samples for training. Note: This figure demonstrates how the data generator creates sequential fall samples for training, selecting every 3rd frame to ensure temporal consistency while maintaining diversity in the input data. Each generated sample contains 10 consecutive frames.

$$\textit{TF-66: PossibleNumberOfStartingFrames} = \left\lfloor \frac{numFrames}{2} \right\rfloor + 1 \tag{5.8}$$

$$\textit{TSF: PossibleNumberOfStartingFrames} = \left\lfloor \frac{numFrames}{2} \right\rfloor \times 3 + 1 \tag{5.9}$$

$$\textit{PossibleUniqueSamples}_{\textit{ForMinorityClass}} = (5.8) \times \textit{Number of Videos}_{\textit{ForMinorityClass}} \tag{5.10}$$

5.3 Experimental Analysis

5.3.1 Environment

All proposed models were developed using Python version 3.10.11 alongside its open-source native libraries and DL frameworks, including Keras with a TensorFlow backend. Model development, training, and testing were conducted on the Cedar core of Compute Canada, utilizing an NVIDIA Tesla K80 GPU (2496 CUDA cores, 12GB VRAM).

5.3.2 Experimental Results

Table 5.1 summarizes the experimental results for the best-performing models across both datasets and under the two data generator approaches. Evaluation metrics are presented for five configurations, categorized by model type and dataset. While dataset- and generator-specific discussions are provided in later sections, this subsection focuses on overarching trends and system performance. The models converged to optimal solutions within 4 to 59 epochs. Models employing attention mechanisms typically required more epochs to converge. The ROC AUC values—a critical metric for thermal fall detection (as noted in Table 2.8)—consistently exceeded 95%, indicating excellent performance. Exceptions are highlighted in the dataset-specific discussions.

The GFLOPs of models incorporating optical flow and self-attention mechanisms were significantly higher than baseline configurations. Adding optical flow approximately doubled the GFLOPs, while self-attention mechanisms quadrupled them. These increases underscore the computational demands of advanced techniques but also highlight their substantial performance benefits. Despite the heightened complexity, per-sample inference times remained practical, generally ranging from 20 to 25 milliseconds. The self-attention model exhibited the longest inference time, reflecting its high GFLOPs. However, all configurations met the real-time processing requirement of 2.5 seconds, dictated by the system's 4fps frame capture rate for 10-frame samples. Real-time performance requires processing each frame within 250 milliseconds to ensure no rapid fall events are missed in overlapping frames. Despite the additional computational burden of optical flow preprocessing, which added approximately 197 milliseconds per sample, models incorporating optical flow achieved a combined inference time of 243 milliseconds—comfortably below the threshold.

One table entry remains blank due to the computational constraints of the self-attention model on the TSF dataset with the balanced data generator. Persistent GPU memory errors and register spillage slowed computations, even after attempts to reduce complexity by adjusting hyperparameters, batch sizes, and image input dimensions. After 15 failed attempts, this configuration was abandoned. These challenges highlight the resource-intensive nature of self-attention mechanisms and caution researchers against their deployment without adequate computational resources. Other

self-attention variants successfully executed with a reduced batch size of 4 samples, but this strategy failed for the final configuration.

Table 5.1: Performance of the proposed 3D CNN models on the validation subset of the TF-66 dataset.
Note: GEN - Dataset generation approach, AUC - Area under the curve of ROC, ACC - Accuracy, FS - F1 Score, EP - Training Epochs, GFLOPS - Giga floating operations per second, PSIT - Per sample inference time in ms

Dataset	Model	GEN	EP	Loss	AUC %	ACC %	FS %	MCC %	GFLOPS	PSIT
TF-66	Baseline	Brute	7	0.349	91.6	85.8	80.2	69.2	37.6	22
TF-66	Baseline	Balance	5	0.381	93.8	86.7	84.9	72.6	37.6	21
TSF	Baseline	Brute	5	0.615	97.4	86.7	90.6	71.9	37.6	22
TSF	Baseline	Balance	11	0.666	96.9	84.7	89.1	68.9	37.6	22
TF-66	CoLSTM + Opt	Brute	18	0.208	96.8	92.4	90.3	84.0	76.3	45
TF-66	CoLSTM + Opt	Balance	22	0.200	97.5	93.8	92.3	87.1	76.3	46
TSF	CoLSTM + Opt	Brute	4	0.376	88.4	81.5	87.7	51.5	76.3	44
TSF	CoLSTM + Opt	Balance	5	0.410	89.7	84.6	89.9	59.2	76.3	46
TF-66	BiCoLSTM + Att	Brute	45	0.226	95.9	91.6	87.9	81.6	38.6	25
TF-66	BiCoLSTM + Att	Balance	26	0.225	97.0	90.4	86.5	79.0	38.6	25
TSF	BiCoLSTM + Att	Brute	59	0.222	97.0	89.6	93.2	72.7	38.6	25
TSF	BiCoLSTM + Att	Balance	46	0.071	99.7	96.9	98.0	90.9	38.6	24
TF-66	CoLSTM + Att	Brute	42	0.205	96.8	93.2	90.6	86.3	38.3	22
TF-66	CoLSTM + Att	Balance	21	0.225	97.4	90.5	89.2	81.0	38.3	22
TSF	CoLSTM + Att	Brute	39	0.267	89.7	90.6	94.4	65.4	38.3	23
TSF	CoLSTM + Att	Balance	24	0.227	96.9	90.6	93.5	76.8	38.3	23
TF-66	Self Att	Brute	32	0.185	97.9	92.4	89.4	83.5	148.7	87
TF-66	Self Att	Balance	10	0.295	94.6	88.9	86.0	76.8	148.7	89
TSF	Self Att	Brute	6	0.402	94.0	82.3	87.4	61.7	148.7	86
TSF	Self Att	Balance	-	-	-	-	-	-	-	-

5.3.3 Quantitative Analysis

Given the inherent differences between the TF-66 and TSF datasets, it is crucial to evaluate model performance on each dataset independently to determine the most effective configurations. This subsection begins with an analysis of model performance on the TF-66 dataset, followed by a discussion of results on the TSF dataset. A comparison of dataset generation approaches will be deferred until after these dataset-specific discussions. Table 5.2 presents the results for the TF-66 dataset, extracted from Table 5.1. All proposed models outperform the baseline models across evaluation metrics. Enhanced models achieve ROC-AUC values of 97% or higher, with accuracies exceeding 90%, F1 scores surpassing 85%, and MCC scores exceeding 80%. Among these, the ConvLSTM + Optical Flow model emerges as the most robust and effective.

Although the self-attention network trained with the brute-force dataset generation approach achieves the highest ROC-AUC, this advantage is marginal (0.4%) compared to the ConvLSTM + Optical Flow model. Moreover, the latter surpasses the self-attention model in accuracy (1.4% higher), F1 score (2.9% higher), and MCC (3.6% higher). When averaging performance across brute-force and balanced data generation approaches, the ConvLSTM + Optical Flow model outperforms the Self-Attention network (97.2% vs. 96.3%), highlighting its superior robustness on the TF-66 dataset.

Both the ConvLSTM + Optical Flow and self-attention models involve significantly higher computational complexity, as indicated by their GFLOPs, yet they remain capable of real-time operation. Notably, the ConvLSTM architecture consistently outperformed Bi-ConvLSTM models under similar conditions. The integration of spatial, temporal, and feature attention mechanisms into RNN-based models also proved effective, with two configurations ranking among the top four performers. Interestingly, combining ConvLSTM architectures with attention mechanisms and optical flow did not yield the anticipated performance improvements. Instead, performance dropped below that of simpler top-performing models. This decline is likely due to overfitting caused by excessive model complexity or suboptimal integration of optical flow data.

The ConvLSTM + Optical Flow model achieves the best balance between maximizing the detection rate of falls—critical for identifying the positive class—and minimizing false positives. Additionally, its MCC score, which holistically evaluates all confusion matrix elements, underscores the model's reliability and effectiveness. Based on these findings, the ConvLSTM + Optical Flow model is identified as the superior configuration for the TF-66 dataset. Further discussion of this model's characteristics and performance will follow in Section 5.4. The models that performed optimally on the TF-66 dataset were subsequently evaluated on the TSF dataset. The filtered results from Table 5.1 are summarized in Table 5.3. Performance on the TSF dataset was less conclusive compared to TF-66, with no single model emerging as a clear overall best. However, the Bi-ConvLSTM + Attention Mechanisms model achieved state-of-the-art results, including an AUC of 99.7%, accuracy and F1 scores exceeding 95%, and an MCC surpassing 90%. These metrics

Table 5.2: The results of the proposed 3D CNN models trained and evaluated on the validation subset of the TF-66 dataset, which was unseen during training
Note: GEN - Dataset generation approach, AUC - Area under the curve of ROC, ACC - Accuracy, FS - F1 Score, EP - Training Epochs, GFLOPS - Giga floating operations per second, PSIT - Per sample inference time in ms, CoLSTM - Convolutional LSTM, Opt - Optical Flow, Att - Attention Mechanisms

Dataset	Model	GEN	EP	Loss	AUC %	ACC %	FS %	MCC %	GFLOPS	PSIT
TF-66	Baseline	Brute	7	0.349	91.6	85.8	80.2	69.2	37.6	22
TF-66	Baseline	Balance	5	0.381	93.8	86.7	84.9	72.6	37.6	21
TF-66	CoLSTM + Opt	Brute	18	0.208	96.8	92.4	90.3	84.0	76.3	45
TF-66	CoLSTM + Opt	Balance	22	0.200	97.5	93.8	92.3	87.1	76.3	46
TF-66	BiCoLSTM + Att	Brute	45	0.226	95.9	91.6	87.9	81.6	38.6	25
TF-66	BiCoLSTM + Att	Balance	26	0.225	97.0	90.4	86.5	79.0	38.6	25
TF-66	CoLSTM + Att	Brute	42	0.205	96.8	93.2	90.6	86.3	38.3	22
TF-66	CoLSTM + Att	Balance	21	0.225	97.4	90.5	89.2	81.0	38.3	22
TF-66	Self Att	Brute	32	0.185	97.9	92.4	89.4	83.5	148.7	87
TF-66	Self Att	Balance	10	0.295	94.6	88.9	86.0	76.8	148.7	89

Table 5.3: The results of the proposed 3D CNN models trained and evaluated on the validation subset of the TSF dataset, which was unseen during training
Note: GEN - Dataset generation approach, AUC - Area under the curve of ROC, ACC - Accuracy, FS - F1 Score, EP - Training Epochs, GFLOPS - Giga floating operations per second, PSIT - Per sample inference time in ms, CoLSTM - Convolutional LSTM, Opt - Optical Flow, Att - Attention Mechanisms

Dataset	Model	GEN	EP	Loss	AUC %	ACC %	FS %	MCC %	GFLOPS	PSIT
TSF	Baseline	Brute	5	0.615	97.4	86.7	90.6	71.9	37.6	22
TSF	Baseline	Balance	11	0.666	96.9	84.7	89.1	68.9	37.6	22
TSF	CoLSTM + Opt	Brute	4	0.376	88.4	81.5	87.7	51.5	76.3	44
TSF	CoLSTM + Opt	Balance	5	0.410	89.7	84.6	89.9	59.2	76.3	46
TSF	BiCoLSTM + Att	Brute	59	0.222	97.0	89.6	93.2	72.7	38.6	25
TSF	BiCoLSTM + Att	Balance	46	0.071	99.7	96.9	98.0	90.9	38.6	24
TSF	CoLSTM + Att	Brute	39	0.267	89.7	90.6	94.4	65.4	38.3	23
TSF	CoLSTM + Att	Balance	24	0.227	96.9	90.6	93.5	76.8	38.3	23
TSF	Self Att	Brute	6	0.402	94.0	82.3	87.4	61.7	148.7	86
TSF	Self Att	Balance	-	-	-	-	-	-	-	-

highlight the model's exceptional effectiveness. While the ConvLSTM + Attention Mechanisms model also performed well, it was outperformed by its bidirectional counterpart.

Interestingly, the model integrating optical flow, which excelled on the TF-66 dataset, exhibited the poorest performance on the TSF dataset. This disparity is likely due to differences in the datasets and the fixed optical flow generation parameters, which were optimized for TF-66 but not for TSF (See Appendix D for information regarding optical flow image generation). The TF-66 dataset comprises thermal-only images capturing heat signatures, while the TSF dataset includes

images illuminated with visible light, revealing environmental details. These differences may have influenced the effectiveness of the optical flow parameters, suggesting that dataset-specific parameter optimization could improve performance. The self-attention mechanism demonstrated strong performance on the TSF dataset but fell short of the Bi-ConvLSTM + Attention Mechanisms model. Additionally, computational resource constraints prevented the final test configuration of the self-attention model from being run, further limiting its utility for this dataset.

A cross-dataset comparison reveals interesting trends. On the TF-66 dataset, ConvLSTM models consistently outperformed other configurations, whereas Bi-ConvLSTM models excelled on the TSF dataset. This divergence is likely attributable to the datasets' differing levels of detail. The privacy-preserving TF-66 dataset, with its lower-resolution thermal-only images, captures less detail and fewer distinguishable features. ConvLSTM models, which emphasize forward temporal dependencies, are better suited to these simpler motion patterns. In contrast, the TSF dataset's visible-light images provide higher detail, including clear outlines and limb movements, which favor the bidirectional processing capabilities of Bi-ConvLSTMs. These models can effectively leverage nuanced temporal relationships along image sequences by analyzing both past and future frames. Each model was trained on both the TF-66 and TSF datasets using the "balanced" and "brute force" data generation methods. The experimental results, summarized in Table 5.4, compare model performance across these methods. The percentage difference equation (5.11) was applied, using the "balanced" method as the baseline. Positive values indicate that the "balanced" method outperformed the "brute force" method, and vice versa.

$$\text{Percentage Difference} = \frac{|\text{Brute\_Result} - \text{Balanced\_Result}|}{\text{Balanced\_Result}} \times 100 \tag{5.11}$$

Analyzing the results reveals several trends. When optical flow was included in the model architecture, the balanced method consistently outperformed the brute force method across both datasets. This can be attributed to the balanced approach's ability to preserve class representation and prevent overfitting, enabling the model to generalize better. Optical flow and thermal frames complement one another by providing distinct yet interconnected information on motion

and temperature. The balanced method's equal class representation ensures these data channels work synergistically without memorization, improving fall detection performance.

Table 5.4: Performance comparison of the balanced and brute data generator approaches on TF-66 and TSF validation subsets

Note: An upwards arrow (↑) indicates the percentage improvement of the balanced data generator approach over the brute approach, while a downwards arrow (↓) indicates a percentage reduction. Key metrics include AUC - Area under the curve, ACC - Accuracy, FS - F1 Score, and MCC - Matthews Correlation Coefficient

Dataset	Model	Loss	AUC %	ACC %	FS %	MCC %
TF-66	Baseline	↓ 9.16	↑ 2.40	↑ 1.05	↑ 5.86	↑ 4.91
TF-66	CoLSTM + Opt	↓ 3.85	↑ 1.52	↑ 1.49	↑ 2.21	↑ 3.69
TF-66	BiCoLSTM + Att	↑ 0.44	↑ 1.15	↓ 1.31	↓ 1.59	↓ 3.19
TF-66	CoLSTM + Att	↓ 9.76	↑ 0.62	↓ 2.90	↓ 1.55	↓ 6.14
TF-66	Self Att	↓ 59.46	↓ 3.37	↓ 3.79	↓ 3.80	↓ 8.02
TSF	Baseline	↓ 8.29	↓ 0.51	↓ 2.31	↓ 1.66	↓ 4.17
TSF	CoLSTM + Opt	↓ 9.04	↑ 1.47	↑ 3.80	↑ 2.51	↑ 14.95
TSF	BiCoLSTM + Att	↑ 68.02	↑ 2.78	↓ 8.15	↓ 5.15	↓ 25.03
TSF	CoLSTM + Att	↑ 14.98	↑ 8.03	-	↓ 0.95	↑ 17.43

In contrast, when optical flow was excluded, the brute force method yielded better results for most metrics across both datasets. This trend highlights the impact of dataset size. Despite TF-66 containing 562 fall samples—substantially more than TSF—both datasets remain small by DL standards, where datasets typically consist of thousands or millions of samples. The brute force approach, by repeatedly exposing the model to the same fall samples within each epoch, acts as a form of oversampling. This reinforcement allows the model to internalize critical patterns, enhancing its ability to detect falls. Notably, the metrics in Table 5.4 were obtained on the validation dataset, meaning overfitting did not impact performance as this validation data was not exposed to the models until after training had been completed. The repeated exposure proved advantageous in this context, enabling the model to learn representative features despite the limited dataset size.

Based on these findings, future research can tailor strategies to the specific context of the dataset. For smaller datasets, a brute force approach might work better, as it emphasizes key patterns essential for effective feature extraction. On the other hand, for larger, more diverse datasets, the balanced method is likely more effective, as it addresses class imbalance and supports better generalization. These observations offer a practical starting point for refining training strategies to suit different dataset characteristics.

5.3.4 Qualitative Analysis

To visually evaluate the performance of the proposed model, specific samples were examined to assess their classification accuracy. Figure 5.3 illustrates 8 consecutive frames and their corresponding optical flow representations, extracted from video `01-Fall-04`, beginning at frame 35. This sample was consistently classified correctly across all tests.

The classification aligns with the expected characteristics of falls in the TF-66 dataset. Between frames 35 and 39, a dense, circular heat signature transitions into a dimmer, elongated heat signature, representing the individual falling to the ground. This elongated signature persists in subsequent frames (frames 40 to 42), indicating that the individual is lying prone rather than bending over. The optical flow data corroborates this behavior, with high optical flow intensities during the fall (frames 35 to 39) and minimal activity in later frames (frames 40 to 42) as the individual remains stationary.

This qualitative analysis underscores the model's capability to effectively integrate spatial and temporal features from thermal imagery and optical flow. By leveraging these complementary modalities, the model demonstrates a robust ability to accurately detect and classify falls. Upon manual examination, the most common misclassifications occurred in fall samples with surrounding thermal noise that diverted attention away from the human subject and onto irrelevant features. An illustrative example of a consistently misclassified sample is presented in Figure 5.4, highlighting several potential factors contributing to this misclassification.

First, frames 32 and 33 contain a non-human heat signature that interferes with optical flow calculations, slightly distorting the appearance of subsequent frames, such as frame 34. When the individual falls, their heat signature closely matches the intensity of residual heat detected in the bottom corners of the image. This residual heat, emitted by a vent during recording, appears brighter than the individual's heat signature. As a result, the individual's signature appears less distinct, and the automatic human detection arrow is mistakenly directed toward the heat source in the corner. This misclassification impacts the model's performance in these specific frames. Additionally, the nature of the fall itself likely contributes to the misclassification. In this instance, the individual transitions from a stationary position to a sideways fall. Unlike falls preceded by active

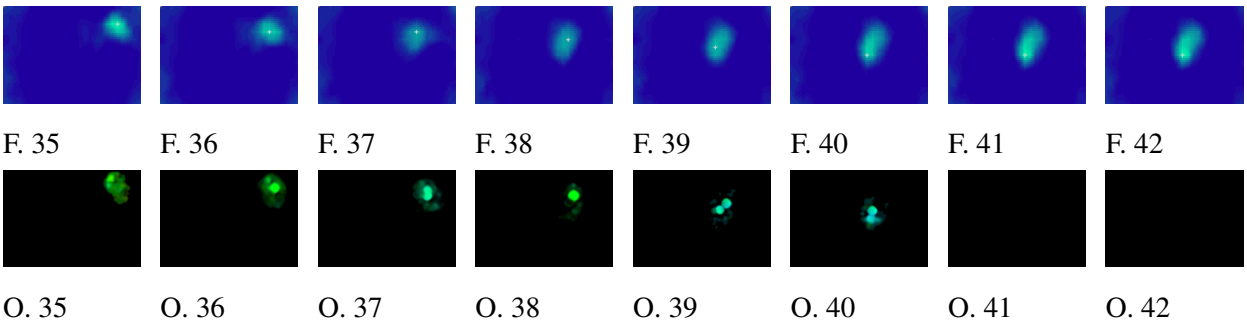


Figure 5.3: Eight consecutive frames from 01-Fall-04 starting at frame 35 in the TF-66 Dataset, including original images (top row) and optical flow data (bottom row).
Note: The video depicts a person walking into the scene, falling onto their stomach, and remaining in a prone position. Optical flow in the second row highlights motion differences between consecutive frames.

motion, this subtle type of fall is harder to detect, increasing the probability of being overlooked by the model.

Lastly, the optical flow output for this sample reveals unexpected motion in frames 37 to 39, even after the individual has fallen and is expected to remain stationary. This unexpected motion contradicts the behavior for a fall event, likely compounding the model's confusion. While such challenges are inevitable in large datasets, they underscore the robustness of the TF-66 dataset. These inherent complexities prevent models from achieving inflated accuracy scores prematurely, encouraging researchers to refine their approaches. Ultimately, this fosters the development of solutions that are more resilient to real-world applications.

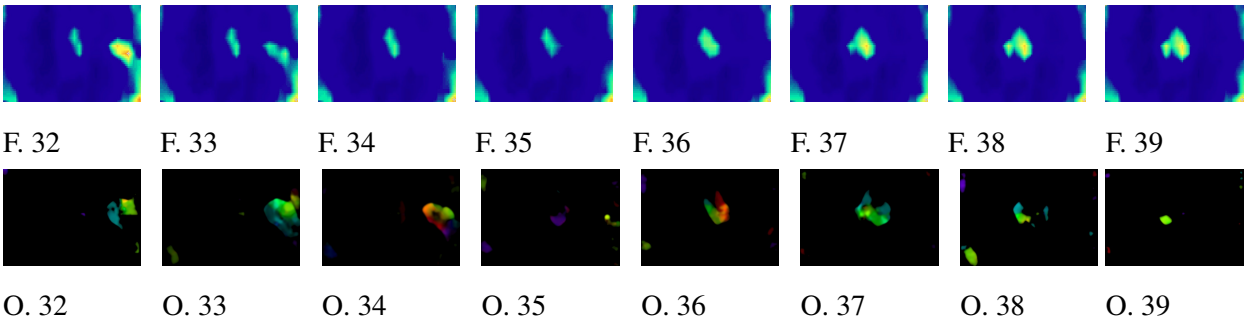


Figure 5.4: Eight consecutive frames from 11-Fall-08 starting at frame 32 in the TF-66 Dataset, including original images (top row) and optical flow data (bottom row).
Note: This video depicts a person standing upright and collapsing down to their side. The optical flow in the second row highlights motion changes between consecutive frames.

5.4 Optimal Models

The most effective model for the TF-66 dataset is the ConvLSTM + Optical Flow model, as shown in Table 5.2. This model builds on the updated baseline architecture, incorporating a Convolutional LSTM layer after the Conv3D layers, along with a generic global attention mechanism connecting to the input features and an additional input channel containing optical flow data between consecutive frames. The specific architecture is detailed in Table 5.5, with a corresponding visual representation in Figure 5.5. The "Layer ID" column in the table aligns with layers in the figure to facilitate understanding.

Table 5.5: Architectural detail of the 3D CNN model that performs best on the TF-66 dataset
Note: The ConvLSTM2D module enables multi-channel inputs, combining thermal frames and their optical flow counterparts to better capture motion dynamics.

Layer ID	Layer Type	Output Dimension
Input	Input Layer	(16, 10, 256, 256, 1)
L1 - Conv1	Conv3D	(16, 10, 256, 256, 32)
L2 - MaxPool2	MaxPooling3D	(16, 10, 128, 128, 32)
L3 - Dropout3	Dropout	(16, 10, 128, 128, 32)
L4 - Conv4	Conv3D	(16, 10, 128, 128, 64)
L5 - MaxPool5	MaxPooling3D	(16, 10, 64, 64, 64)
L6 - Dropout6	Dropout	(16, 10, 64, 64, 64)
L7 - Conv7	Conv3D	(16, 10, 64, 64, 128)
L8 - MaxPool8	MaxPooling3D	(16, 10, 32, 32, 128)
L9 - Dropout9	Dropout	(16, 10, 32, 32, 128)
L10 - Reshape10	Reshape	(16, 10, 32, 32, 128)
L11 - ConvLSTM2D11	Forward ConvLSTM2D	(16, 10, 32, 32, 64)
L12 - Reshape12	Reshape12	(10240, 64)
L13 - GeneralAttention13	General Attention13	(16, 128)
L14 - Dense14	Dense14	(16, 64)
L15 - Dropout15	Dropout15	(16, 64)
Output - Dense16	Dense16	(16, 1)

Total number of trainable parameters: 725,346 (2.77 MB); Activation function: L1, L5, L9: LeakyReLU;
Attention Layers: Custom activations; Output: sigmoid; Kernel size: (3,3,3) for Conv3D operations, (1,2,2) for MaxPooling3D layers, (2,2) for Conv2D layers; Padding: Same padding is always used;
Dropout rate: L3, L7, L11: 0.25; L20: 0.5; Learning rate: 0.001; Optimizer: Adam
Number of Epochs: 75; Batch size: 16; Loss Function: Binary cross-entropy / log loss

Meanwhile, the best-performing model on the TSF dataset is the BiConvLSTM + Attention Mechanisms model, described in Table 5.6 and illustrated in Figure 5.6. This model incorporates a BiConvLSTM layer after the Conv3D layers and employs multiple attention mechanisms. In addition to the generic global attention mechanism, it includes spatial, temporal, and feature attention

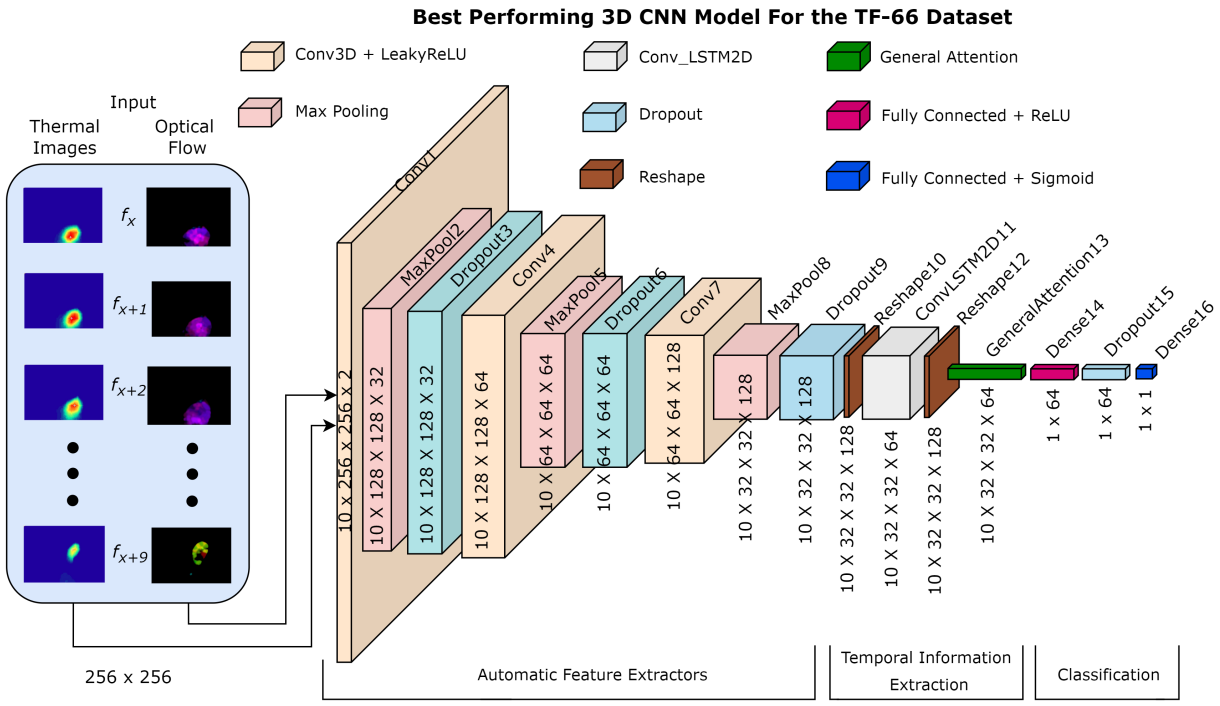


Figure 5.5: The architecture of the best performing 3D CNN model introduced in Chapter 4.
Note: This model improves temporal processing by integrating optical flow as an input and adding a ConvLSTM module after the final Conv3D layer.

mechanisms applied after the first, second, and third Conv3D layers, respectively. These attention mechanisms several purposes:

- **Global Attention:** Operates across all input features, enhancing feature-level focus.
- **Spatial Attention:** Highlights specific regions within an image or grid.
- **Temporal Attention:** Identifies the most relevant frames in a sequence.
- **Feature Attention:** Emphasizes important channel features across the input tensor.

The combination of these specialized mechanisms allows the model to extract deeper and more nuanced features, contributing to state-of-the-art performance on the TSF dataset. This robust integration of diverse attention mechanisms enables the model to excel in vision-based fall detection, achieving unprecedented levels of accuracy and generalizability.

Table 5.6: Architectural detail of the 3D CNN model that performs best on the TSF dataset
Note: The addition of attention mechanisms and a BiConvLSTM2D module captures temporal data more effectively. The BiConvLSTM2D layer combines forward and backward ConvLSTM2D outputs through concatenation to achieve bidirectional temporal feature extraction.

Layer ID	Layer Type	Output Dimension
Input	Input Layer	(16, 10, 256, 256, 1)
L1 - Conv1	Conv3D	(16, 10, 256, 256, 32)
L2 - MaxPool2	MaxPooling3D	(16, 10, 128, 128, 32)
L3 - Dropout3	Dropout	(16, 10, 128, 128, 32)
L4 - SpatialAttention2	Spatial Attention	(16, 10, 128, 128, 32)
L5 - Conv5	Conv3D	(16, 10, 128, 128, 64)
L6 - MaxPool6	MaxPooling3D	(16, 10, 64, 64, 64)
L7 - Dropout7	Dropout	(16, 10, 64, 64, 64)
L8 - TemporalAttention8	Temporal Attention	(16, 10, 64, 64, 64)
L9 - Conv9	Conv3D	(16, 10, 64, 64, 128)
L10 - MaxPool10	MaxPooling3D	(16, 10, 32, 32, 128)
L11 - Dropout11	Dropout	(16, 10, 32, 32, 128)
L12 - FeatureAttention12	Feature-Based Attention	(16, 10, 32, 32, 128)
L13 - Reshape13	Reshape	(16, 10, 32, 32, 128)
L14 - BiConvLSTM2D14	BiConvLSTM2D	(16, 10, 32, 32, 128)
L15 - Concatenate15	Concatenate	(16, 10, 32, 32, 128)
L16 - Reshape16	Reshape	(16, 10, 128)
L17 - GeneralAttention17	General Attention	(16, 128)
L18 - Dense18	Dense	(16, 64)
L19 - Dropout19	Dropout	(16, 64)
Output - Dense20	Dense	(16, 1)

Total number of trainable parameters: 1,172,422 and memory size of 4.47 MB;
Activation function: (L1, L5, L9)→LeakyReLU (alpha = 0.1), spatial and feature attention → sigmoid, temporal attention→softmax, Output→sigmoid; Reduction ratio: 32 for feature attention;
Kernel size: (3,3,3)→Conv3D operations, (1,2,2)→MaxPooling3D layers, (2,2)→BiConvLSTM2D layers;
Padding: Same padding is always used; Dropout rate: (L3, L7, L11)→0.25, L19→0.5; Learning rate: 0.001;
Optimizer: Adam; Number of Epochs: 75; Batch size: 16; Loss Function: Binary cross-entropy / log loss

5.5 Real-World Implications of the Existing Methods

The proposed BiConvLSTM + Attention Mechanism model has set a new benchmark, achieving an ROC-AUC of 99.7% on the TSF dataset. Table 5.7 provides a comparative analysis, clearly illustrating its superior performance. The primary drivers of this improvement are the integration of multiple specialized attention mechanisms and the inclusion of a Bi-LSTM module with a generic attention mechanism connecting globally to all input features. These attention mechanisms enable the model to identify nuanced features and patterns that might otherwise be overlooked. Meanwhile, this updated model has also improved upon the state-of-the-art performance as initially introduced in Chapter 4. While the TF-66 dataset has yet to be evaluated by external researchers,

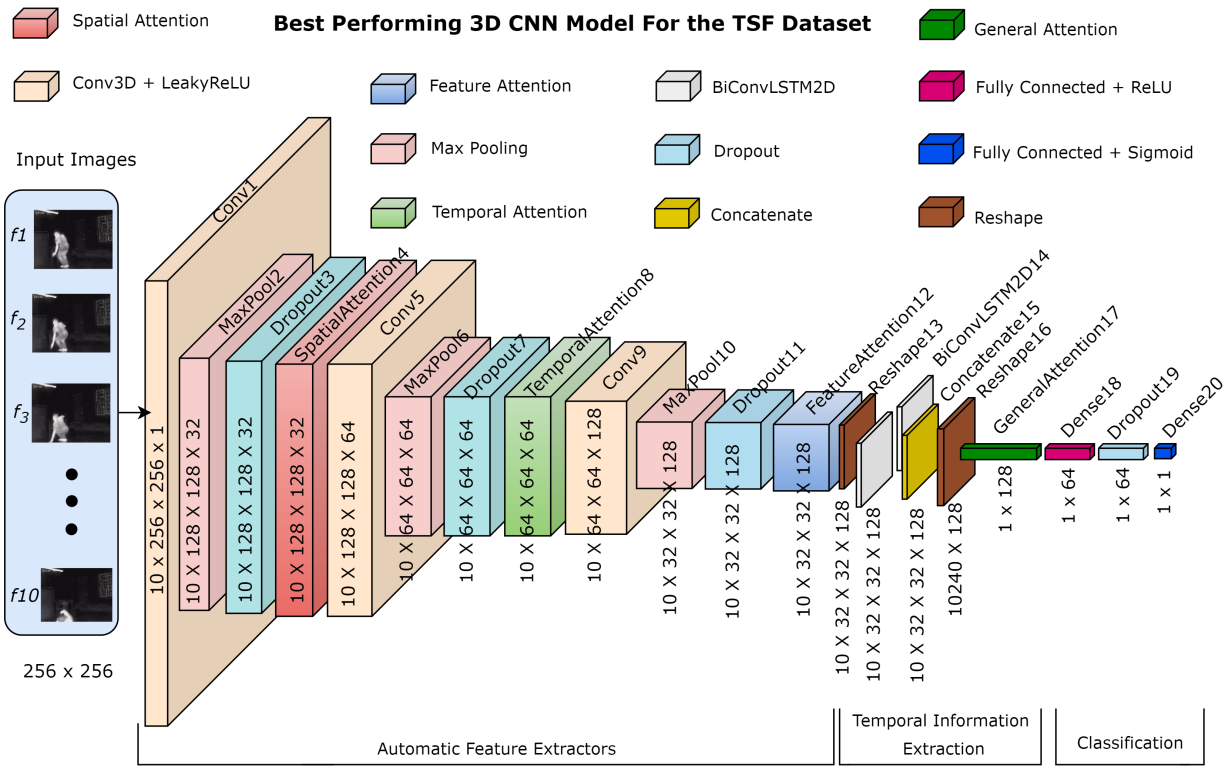


Figure 5.6: The architecture of the best performing 3D CNN model on the TSF dataset [2].
Note: An illustration of the enhanced model. This model extends a baseline 3D CNN by adding a spatial attention layer, temporal attention layer, and feature attention layer after the first three Conv3D blocks, respectively. Following the feature attention layer, a BiConvLSTM2D module is included to enhance temporal processing. Finally, a general attention layer is applied before classification.

and therefore lacks comparative performance metrics from independent studies, the initial baseline results have already been surpassed, as demonstrated in Table 5.8, with the 10, 9, and 8 foot room subsets all surpassing their baseline metric value across all metrics. The performance of the hospital subset significantly decreased, but this is likely due to the low number of samples within this subset, which allow a few incorrect classificiations to skew the data. The senior subset only performed slightly worse, however the entire dataset as a whole performed much better, increasing to an ROC-AUC value of 97.4%. This advancement establishes another milestone for future researchers to aim to exceed, driving further progress in thermal fall detection research.

Table 5.7: Performance comparison of various models on the TSF dataset
Note: Note: AUC - Area under the curve of ROC, GFLOPS - giga floating-point operations per second, and PSIT - Per sample inference time in ms. The mean (μ) and standard deviation (σ) indicate the anomaly score of the reconstruction error across frames

Model	AUC %	% Improvement	GFLOPS	PSIT	Year
DAE [116]	64	↓ 14.67	-	-	2018
CAE Deconv. [116]	75	Baseline	-	-	2018
ConvLSTM-AE (μ) [116]	76	↑ 1.33	-	-	2018
ConvLSTM-AE (σ) [116]	83	↑ 10.67	-	-	2018
CLSTMAE [118]	83	↑ 10.67	-	-	2020
SRAE [118]	97	↑ 29.33	-	-	2020
DSTCAE-C3D (μ) [76]	93	↑ 24.00	-	-	2020
DSTCAE-C3D (σ) [76]	97	↑ 29.33	-	-	2020
Adversarial learning (μ) [75]	95	↑ 26.67	-	-	2021
Adversarial learning (σ) [75]	95	↑ 26.67	-	-	2021
Fusion-Diff-ROI-3DCAE (μ) [70]	93	↑ 24.00	-	-	2021
Fusion-Diff-ROI-3DCAE (σ) [70]	93	↑ 24.00	-	-	2021
3D CNN (Chapter 3)	79	↑ 5.33	17.7	-	2023
AE (Chapter 3)	74	↓ 1.33	4.03	-	2023
3D CNN-AE (Chapter 3)	83	↑ 10.67	21.76	-	2023
Proposed Model	99.7	↑ 32.93	38.6	24	2024

Table 5.8: Performance of the baseline and the proposed model on the various subsets of TF-66
Note: OF - Optical flow, Subset - TF-66 subsets (e.g., 10ft, 9ft, 8ft rooms), Hosp - Hospital, Snr - Senior, AUC - Area under the curve of ROC, ACC - Accuracy, FS - F1 Score, GFLOPS - Giga floating operations per second, % I - percentage improvement

Model	Subset	AUC %	% I.	ACC %	% I.	FS %	% I.	MCC %	% I.
Baseline	TF-66	92.9	-	84.5	-	80.4	-	67.6	-
ConvLSTM + OF	TF-66	97.4	↑ 4.38	90.5	↑ 7.10	89.2	↑ 10.95	81.0	↑ 19.82
Baseline	10ft	95.2	-	90.7	-	90.6	-	82.1	-
ConvLSTM + OF	10ft	98.4	↑ 3.36	93.6	↑ 3.20	92.1	↑ 1.66	87.2	↑ 6.21
Baseline	9ft	90.2	-	75.6	-	79.7	-	55.1	-
ConvLSTM + OF	9ft	99.1	↑ 9.87	96.7	↑ 27.91	97.0	↑ 21.71	93.6	↑ 69.87
Baseline	8ft	89.6	-	81.5	-	73.2	-	59.5	-
ConvLSTM + OF	8ft	93.2	↑ 4.02	86.7	↑ 6.38	79.2	↑ 8.20	69.4	↑ 16.64
Baseline	Hosp.	98.5	-	93.7	-	83.4	-	79.7	-
ConvLSTM + OF	Hosp.	88.8	↓ 9.85	88.0	↓ 6.08	53.6	↓ 35.73	47.8	↓ 40.03
Baseline	Snr.	99.4	-	97.3	-	91.7	-	90.1	-
ConvLSTM + OF	Snr.	95.2	↓ 4.23	95.6	↓ 1.75	84.0	↓ 8.40	81.6	↓ 9.43

5.6 Chapter Summary

This chapter presented an exploration of various advanced methods, starting with an optimization of the baseline 3D CNN model. The ConvLSTM + Optical Flow model emerged as the optimal architecture for the TF-66 dataset, leveraging spatiotemporal clues and motion data to achieve

robust and generalizable results. Similarly, the BiConvLSTM + Attention Mechanisms set a new benchmark on the TSF dataset, attaining a record ROC-AUC of 99.7%. It also investigated the impact of data generation approaches, demonstrating that the brute force method outperformed the balanced approach on these smaller datasets, while balanced generation proved more effective when integrating multiple data channels such as optical flow. Qualitative analyses highlighted the model's ability to handle complex fall patterns. Through cross-dataset evaluations, this chapter demonstrated the proposed models' ability to generalize and adapt to diverse data characteristics. The findings highlight meaningful progress in fall detection, providing benchmark models and datasets that can guide future research in the field.

Chapter 6

Conclusion

This thesis presents a comprehensive framework for advancing fall detection research, addressing critical challenges in the field such as the lack of robust datasets, generalizability, and real-world applicability. By leveraging thermal imaging and innovative deep learning techniques, this work bridges the gap between controlled lab settings and practical deployment, ultimately contributing to the development of accurate, privacy-preserving solutions, that can help the tens of millions of seniors that fall annually worldwide.

The research begins with an exploration of hybrid architectures that combine supervised and unsupervised learning paradigms, revealing key limitations in existing datasets and motivating the creation of TF-66—a diverse and representative thermal fall detection dataset. TF-66 serves as a cornerstone for advancing research, enabling fair model comparisons and meaningful progress toward real-world solutions. Building on this foundation, the thesis develops state-of-the-art models that integrate ConvLSTMs, optical flow, and attention mechanisms, achieving exceptional performance across benchmark datasets.

Beyond model design, this work introduces tools and methodologies, including a brute-force data generation approach and flexible data generators, that provide researchers with versatile solutions for constrained datasets. By making TF-66 publicly available, this thesis encourages collaboration, fostering innovation and establishing a standard for future advancements in fall detection research.

Future work will be focused on expanding the TF-66 dataset to include underrepresented scenarios, such as stair-related falls, multi-person environments, and bathroom settings, to further enhance its applicability. Refining data generation parameters, such as `start-min`, `start-max`, and the number of frames per sample, could also improve model performance and adaptability. These efforts will ensure that TF-66 remains a dynamic and relevant benchmark for advancing fall detection research.

In conclusion, this thesis highlights the transformative impact of robust datasets and systematic experimentation in advancing fall detection research. From the foundational exploration of hybrid architectures in Chapter 3 to the development of the TF-66 dataset in Chapter 4 and the state-of-the-art model proposed in Chapter 5, this work provides a cohesive framework for transitioning fall detection from controlled lab settings to practical real-world applications.

The research journey began with defining the requirements for an effective fall detection system, identifying gaps in existing datasets and methodologies. This led to the creation of TF-66, a dataset designed to reflect real-world fall distributions while maintaining robustness and privacy. Building on this foundation, the proposed model not only sets a new state-of-the-art in performance but also operates in real time, achieving the dual objectives of accuracy and practicality.

The contributions of this thesis address current limitations in the field and lay the groundwork for scalable, privacy-preserving solutions that can make meaningful impacts on real-world systems. By aligning research advancements with real-world applicability, this work provides a valuable step forward in the development of life-saving fall detection technologies.

Bibliography

[1] J. R. Crenshaw, K. A. Bernhardt, S. J. Achenbach, E. J. Atkinson, S. Khosla, K. R. Kaufman, and S. Amin, "The circumstances, orientations, and impact locations of falls in community-dwelling older women," *Archives of gerontology and geriatrics*, vol. 73, pp. 240–247, 2017.

[2] S. Khan, "Thermal simulated fall," Available from: https://github.com/ivineetm007/Fall-detection, 2018.

[3] X. Yu, T. Ma, J. Jang, and S. Xiong, "Data augmentation to address various rotation errors of wearable sensors for robust pre-impact fall detection," *IEEE journal of biomedical and health informatics*, vol. 27, no. 5, pp. 2197–2207, 2022.

[4] US Department of Health and Human Services and others, "Falls and fractures in older adults: Causes and prevention," 2023.

[5] X. Chen, J. Yan, S. Qin, P. Li, S. Ning, and Y. Liu, "Fall detection method based on a human electrostatic field and vmd-ecanet architecture," *IEEE Journal of Biomedical and Health Informatics*, 2024.

[6] P. H. A. of Canada, "Seniors' falls in canada: Second report," https://www.canada.ca/en/public-health/services/health-promotion/aging-seniors/publications/publications-general-public/seniors-falls-canada-second-report.html, 2023.

[7] E. Alam, A. Sufian, P. Dutta, and M. Leo, "Vision-based human fall detection systems using deep learning: A review," *Computers in biology and medicine*, vol. 146, p. 105626, 2022.

[8] A. Naser, A. Lotfi, M. D. Mwanje, and J. Zhong, "Privacy-preserving, thermal vision with human in the loop fall detection alert system," *IEEE Transactions on Human-Machine Systems*, vol. 53, no. 1, pp. 164–175, 2022.

[9] D. G. Arnaoutoglou, D. Dedemadis, A.-A. Kyriakou, S. Katsimentes, A. Grekidis, D. Menychtas, N. Aggelousis, G. C. Sirakoulis, and G. A. Kyriacou, "Acceleration-based low-cost cw radar system for real-time elderly fall detection," *IEEE Journal of Electromagnetics, RF and Microwaves in Medicine and Biology*, 2024.

[10] J. Brieva, H. Ponce, E. Moya-Albor, and L. Martínez-Villaseñor, "An intelligent human fall detection system using a vision-based strategy," in *2019 IEEE 14th International Symposium on Autonomous Decentralized System (ISADS)*. IEEE, 2019, pp. 1–5.

[11] S. Denkovski, S. S. Khan, and A. Mihailidis, "Temporal shift-multi-objective loss function for improved anomaly fall detection," in *Asian Conference on Machine Learning*. PMLR, 2024, pp. 295–310.

[12] Y.-W. Bai, S.-C. Wu, and C. H. Yu, "Recognition of direction of fall by smartphone," in *2013 26th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, 2013, pp. 1–6.

[13] S. Chaudhuri, L. Kneale, T. Le, E. Phelan, D. Rosenberg, H. Thompson, and G. Demiris, "Older adults' perceptions of fall detection devices," *Journal of applied gerontology*, vol. 36, no. 8, pp. 915–930, 2017.

[14] M. M. Hasan, M. S. Islam, and S. Abdullah, "Robust pose-based human fall detection using recurrent neural network," in *2019 IEEE International Conference on Robotics, Automation, Artificial-intelligence and Internet-of-Things (RAAICON)*. IEEE, 2019, pp. 48–51.

[15] X. Wang, J. Ellul, and G. Azzopardi, "Elderly fall detection systems: A literature survey," *Frontiers in Robotics and AI*, vol. 7, p. 71, 2020.

[16] P. Wang, Q. Li, P. Yin, Z. Wang, Y. Ling, R. Gravina, and Y. Li, "A convolution neural network approach for fall detection based on adaptive channel selection of uwb radar signals," *Neural Computing and Applications*, vol. 35, no. 22, pp. 15 967–15 980, 2023.

[17] M. Farsi, "Application of ensemble rnn deep neural network to the fall detection through iot environment," *Alexandria Engineering Journal*, vol. 60, no. 1, pp. 199–211, 2021.

[18] S. Chaudhuri, "Examining the feasibility and acceptability of a fall detection device," PhD dissertation, University of Washington, Washington, 2015.

[19] A. Mukherjee and Z. Zhang, "Multisense: A highly reliable wearable-free human fall detection systems." in *SENSORNETS*, 2020, pp. 29–40.

[20] F. Riquelme, C. Espinoza, T. Rodenas, J.-G. Minonzio, and C. Taramasco, "ehomeseniors dataset: An infrared thermal sensor dataset for automatic fall detection research," *Sensors*, vol. 19, no. 20, p. 4565, 2019.

[21] N. T. Newaz and E. Hanada, "The methods of fall detection: A literature review," *Sensors*, vol. 23, no. 11, p. 5212, 2023.

[22] U. Asif, B. Mashford, S. Von Cavallar, S. Yohanandan, S. Roy, J. Tang, and S. Harrer, "Privacy preserving human fall detection using video data," in *Machine Learning for Health Workshop*. PMLR, 2020, pp. 39–51.

[23] L. Abou, A. Fliflet, L. Hawari, P. Presti, J. J. Sosnoff, H. P. Mahajan, M. L. Frechette, and L. A. Rice, "Sensitivity of apple watch fall detection feature among wheelchair users," *Assistive technology*, vol. 34, no. 5, pp. 619–625, 2022.

[24] C.-P. Liu, J.-H. Li, E.-P. Chu, C.-Y. Hsieh, K.-C. Liu, C.-T. Chan, and Y. Tsao, "Deep learning-based fall detection algorithm using ensemble model of coarse-fine cnn and gru networks," in *2023 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*. IEEE, 2023, pp. 1–5.

[25] D. Yacchirema, J. S. de Puga, C. Palau, and M. Esteve, "Fall detection system for elderly people using iot and ensemble machine learning algorithm," *Personal and Ubiquitous Computing*, vol. 23, no. 5, pp. 801–817, 2019.

[26] J. Fleming and C. Brayne, "Inability to get up after falling, subsequent time on floor, and summoning help: prospective cohort study in people over 90," *Bmj*, vol. 337, 2008.

[27] J. Rafferty, J. Medina-Quero, S. Quinn, C. Saunders, I. Ekerete, C. Nugent, J. Synnott, and M. Garcia-Constantino, "Thermal vision based fall detection via logical and data driven processes," in *2019 IEEE International Conference on Big Data, Cloud Computing, Data Science & Engineering (BCD)*. IEEE, 2019, pp. 35–40.

[28] R. Egawa, A. S. M. Miah, K. Hirooka, Y. Tomioka, and J. Shin, "Dynamic fall detection using graph-based spatial temporal convolution and attention network," *Electronics*, vol. 12, no. 15, p. 3234, 2023.

[29] P. H. A. of Canada, "Seniors' falls in canada: Infographic," https://www.canada.ca/en/public-health/services/publications/healthy-living/seniors-falls-canada-second-report/seniors-falls-canada-infographic.html, 2023.

[30] A. Núñez-Marcos, G. Azkune, and I. Arganda-Carreras, "Vision-based fall detection with convolutional neural networks," *Wireless communications and mobile computing*, vol. 2017, no. 1, p. 9474806, 2017.

[31] Y. W. K. Zoetgnande and J.-L. Dillenseger, "A generic interpretable fall detection framework based on low-resolution thermal images," 2022.

[32] L. Wang, M. Peng, and Q. Zhou, "Pre-impact fall detection based on multi-source cnn ensemble," *IEEE Sensors Journal*, vol. 20, no. 10, pp. 5442–5451, 2020.

[33] S. K. Gharghan and H. A. Hashim, "A comprehensive review of elderly fall detection using wireless communication and artificial intelligence techniques," *Measurement*, p. 114186, 2024.

[34] D. Zhao, T. Song, J. Gao, D. Li, and Y. Niu, "Yolo-fall: a novel convolutional neural network model for fall detection in open spaces," *IEEE Access*, 2024.

[35] J. R. Burwinkel, B. Xu, and J. Crukley, "Preliminary examination of the accuracy of a fall detection device embedded into hearing instruments," *Journal of the American Academy of Audiology*, vol. 31, no. 06, pp. 393–403, 2020.

[36] M. E. Tinetti and C. Kumar, "The patient who falls:"it's always a trade-off"," *The Journal of the American Medical Association*, vol. 303, no. 3, pp. 258–266, 2010.

[37] N. Noury, P. Barralon, G. Virone, P. Boissy, M. Hamel, and P. Rumeau, "A smart sensor based on rules and its evaluation in daily routines," in *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 4. IEEE, 2003, pp. 3286–3289.

[38] N. Noury, A. Fleury, P. Rumeau, A. K. Bourke, G. Laighin, V. Rialle, and J.-E. Lundy, "Fall detection-principles and methods," in *2007 29th annual international conference of the IEEE engineering in medicine and biology society*. IEEE, 2007, pp. 1663–1666.

[39] S. Denkovski, S. S. Khan, B. Malamis, S. Y. Moon, B. Ye, and A. Mihailidis, "Multi visual modality fall detection dataset," *IEEE Access*, vol. 10, pp. 106 422–106 435, 2022.

[40] F. Shu and J. Shu, "An eight-camera fall detection system using human fall pattern recognition via machine learning by a low-cost android box," *Scientific reports*, vol. 11, no. 1, p. 2471, 2021.

[41] S. T. Haque, M. Debnath, A. Yasmin, T. Mahmud, and A. H. H. Ngu, "Experimental study of long short-term memory and transformer models for fall detection on smartwatches," *Sensors*, vol. 24, no. 19, p. 6235, 2024.

[42] C.-C. Chang, Y.-C. Chen, B.-H. Sieh, and Y.-M. Ooi, "A distributed fall detection architecture using ensemble learning," in *2021 IEEE 4th International Conference on Knowledge Innovation and Invention (ICKII)*. IEEE, 2021, pp. 81–84.

[43] L. Quagliarella, N. Sasanelli, and G. Belgiovine, "An interactive fall and loss of consciousness detector system," *Gait & posture*, vol. 28, no. 4, pp. 699–702, 2008.

[44] B. Mirmahboub, S. Samavi, N. Karimi, and S. Shirani, "Automatic monocular system for human fall detection based on variations in silhouette area," *IEEE transactions on biomedical engineering*, vol. 60, no. 2, pp. 427–436, 2012.

[45] C. A. Otto and X. Chen, "Automated fall detection: saving senior lives one fall at a time." *Caring: National Association for Home Care magazine*, vol. 28, no. 3, pp. 44–46, 2009.

[46] M. R. Narayanan, S. R. Lord, M. M. Budge, B. G. Celler, and N. H. Lovell, "Falls management: detection and prevention, using a waist-mounted triaxial accelerometer," in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2007, pp. 4037–4040.

[47] M. Allen and O. Pierce, "Knee replacement device unapproved, but used in surgery," *The New York Times*, 2015.

[48] L. Liu, M. Popescu, M. Skubic, M. Rantz, T. Yardibi, and P. Cuddihy, "Automatic fall detection based on doppler radar motion signature," in *2011 5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops*. IEEE, 2011, pp. 222–225.

[49] Y. Li, K. Ho, and M. Popescu, "A microphone array system for automatic fall detection," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 5, pp. 1291–1301, 2012.

[50] X. Zhou, L.-C. Qian, P.-J. You, Z.-G. Ding, and Y.-Q. Han, "Fall detection using convolutional neural network with multi-sensor fusion," in *2018 IEEE international conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2018, pp. 1–5.

[51] Y. Li, K. Ho, and M. Popescu, "Efficient source separation algorithms for acoustic fall detection using a microsoft kinect," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 3, pp. 745–755, 2013.

[52] D. Zhang, X. Zhang, S. Li, Y. Xie, Y. Li, X. Wang, and D. Zhang, "Lt-fall: The design and implementation of a life-threatening fall detection and alarming system," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 7, no. 1, pp. 1–24, 2023.

[53] R. Igual, C. Medrano, and I. Plaza, "Challenges, issues and trends in fall detection systems," *Biomedical engineering online*, vol. 12, no. 1, p. 66, 2013.

[54] G. Zhao, Z. Mei, D. Liang, K. Ivanov, Y. Guo, Y. Wang, and L. Wang, "Exploration and implementation of a pre-impact fall recognition method based on an inertial body sensor network," *Sensors*, vol. 12, no. 11, pp. 15 338–15 355, 2012.

[55] L. Martínez-Villaseñor, H. Ponce, J. Brieva, E. Moya-Albor, J. Núñez-Martínez, and C. Peñafort-Asturiano, "Up-fall detection dataset: A multimodal approach," *Sensors*, vol. 19, no. 9, p. 1988, 2019.

[56] M. Brenner, N. H. Reyes, T. Susnjak, and A. L. Barczak, "Rgb-d and thermal sensor fusion: a systematic literature review," *IEEE Access*, 2023.

[57] L. Wu, C. Huang, S. Zhao, J. Li, J. Zhao, Z. Cui, Z. Yu, Y. Xu, and M. Zhang, "Robust fall detection in video surveillance based on weakly supervised learning," *Neural networks*, vol. 163, pp. 286–297, 2023.

[58] A. Purwar and I. Chawla, "A systematic review on fall detection systems for elderly healthcare," *Multimedia Tools and Applications*, vol. 83, pp. 43 277–43 302, 2024.

[59] S. Mobsite, N. Alaoui, M. Boulmalf, and M. Ghogho, "Semantic segmentation-based system for fall detection and post-fall posture classification," *Engineering Applications of Artificial Intelligence*, vol. 117, p. 105616, 2023.

[60] Y. Chu, K. Cumanan, S. K. Sankarpandi, S. Smith, and O. A. Dobre, "Deep learning-based fall detection using wifi channel state information," *IEEE Access*, vol. 11, pp. 83 763–83 780, 2023.

[61] C. Su, J. Wei, D. Lin, L. Kong, and Y. L. Guan, "A novel model for fall detection and action recognition combined lightweight 3d-cnn and convolutional lstm networks," *Pattern Analysis and Applications*, vol. 27, no. 1, p. 3, 2024.

[62] S. Tateno, F. Meng, R. Qian, and T. Li, "Human motion detection based on low resolution infrared array sensor," in *2020 59th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, 2020, pp. 1016–1021.

[63] F. Sultana, A. Sufian, and P. Dutta, "Advancements in image classification using convolutional neural network," in *2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*. IEEE, 2018, pp. 122–129.

[64] T. Alanazi and G. Muhammad, "Human fall detection using 3d multi-stream convolutional neural networks with fusion," *Diagnostics*, vol. 12, no. 12, p. 3060, 2022.

[65] S. T. Londei, J. Rousseau, F. Ducharme, A. St-Arnaud, J. Meunier, J. Saint-Arnaud, and F. Giroux, "An intelligent videomonitoring system for fall detection at home: perceptions of elderly people," *Journal of telemedicine and telecare*, vol. 15, no. 8, pp. 383–390, 2009.

[66] J. Klenk, C. Becker, F. Lieken, S. Nicolai, W. Maetzler, W. Alt, W. Zijlstra, J. M. Hausdorff, R. Van Lummel, L. Chiari *et al.*, "Comparison of acceleration signals of simulated and real-world backward falls," *Medical engineering & physics*, vol. 33, no. 3, pp. 368–373, 2011.

[67] M. J. Wilding, L. Seegert, S. Rupcic, M. Griffin, S. Kachnowski, and S. Parasuraman, "Falling short: recruiting elderly individuals for a fall study," *Ageing research reviews*, vol. 12, no. 2, pp. 552–560, 2013.

[68] F. Bagala, C. Becker, A. Cappello, L. Chiari, K. Aminian, J. M. Hausdorff, W. Zijlstra, and J. Klenk, "Evaluation of accelerometer-based fall detection algorithms on real-world falls," *PloS one*, vol. 7, no. 5, p. e37062, 2012.

[69] J. Hill, H. Bird, and S. Johnson, "Effect of patient education on adherence to drug treatment for rheumatoid arthritis: a randomised controlled trial," *Annals of the rheumatic diseases*, vol. 60, no. 9, pp. 869–875, 2001.

[70] V. Mehta, A. Dhall, S. Pal, and S. S. Khan, "Motion and region aware adversarial learning for fall detection with thermal imaging," in *2020 25th international conference on pattern recognition (ICPR)*. IEEE, 2021, pp. 6321–6328.

[71] F. X. Gaya-Morey, C. Manresa-Yee, and J. M. Buades-Rubio, "Deep learning for computer vision based activity recognition and fall detection of the elderly: a systematic review," *Applied Intelligence*, vol. 54, no. 19, pp. 8982–9007, 2024.

[72] S. Chaudhuri, H. Thompson, and G. Demiris, "Fall detection devices and their use with older adults: a systematic review," *Journal of geriatric physical therapy*, vol. 37, no. 4, pp. 178–196, 2014.

[73] G. Ben-Sadoun, E. Michel, C. Annweiler, and G. Sacco, "Human fall detection using passive infrared sensors with low resolution: a systematic review," *Clinical interventions in aging*, pp. 35–53, 2022.

[74] K. D. Bhavani and M. F. Ukrit, "Human fall detection using gaussian mixture model and fall motion mixture model," in *2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA)*. IEEE, 2023, pp. 1814–1818.

[75] S. S. Khan, J. Nogas, and A. Mihailidis, "Spatio-temporal adversarial learning for detecting unseen falls," *Pattern Analysis and Applications*, vol. 24, no. 1, pp. 381–391, 2021.

[76] J. Nogas, S. S. Khan, and A. Mihailidis, "Deepfall: Non-invasive fall detection with deep spatio-temporal convolutional autoencoders," *Journal of Healthcare Informatics Research*, vol. 4, no. 1, pp. 50–70, 2020.

[77] T. Xu, Y. Zhou, and J. Zhu, "New advances and challenges of fall detection systems: A survey," *Applied Sciences*, vol. 8, no. 3, p. 418, 2018.

[78] A. Rezaei, M. C. Stevens, A. Argha, A. Mascheroni, A. Puiatti, and N. H. Lovell, "An unobtrusive human activity recognition system using low resolution thermal sensors, machine and deep learning," *IEEE Transactions on Biomedical Engineering*, vol. 70, no. 1, pp. 115–124, 2023.

[79] C. Ma, A. Shimada, H. Uchiyama, H. Nagahara, and R.-i. Taniguchi, "Fall detection using optical level anonymous image sensing system," *Optics & Laser Technology*, vol. 110, pp. 44–61, 2019.

[80] N.-B. Nguyen, K. Chandrasegaran, M. Abdollahzadeh, and N.-M. Cheung, "Re-thinking model inversion attacks against deep neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 16 384–16 393.

[81] Y. Kong, J. Huang, S. Huang, Z. Wei, and S. Wang, "Learning spatiotemporal representations for human fall detection in surveillance video," *Journal of Visual Communication and Image Representation*, vol. 59, pp. 215–230, 2019.

[82] G. Demiris, D. P. Oliver, J. Giger, M. Skubic, and M. Rantz, "Older adults' privacy considerations for vision based recognition methods of eldercare applications," *Technology and Health Care*, vol. 17, no. 1, pp. 41–48, 2009.

[83] K. Wild, L. Boise, J. Lundell, and A. Foucek, "Unobtrusive in-home monitoring of cognitive and physical health: Reactions and perceptions of older adults," *Journal of applied gerontology*, vol. 27, no. 2, pp. 181–200, 2008.

[84] C. He, S. Liu, G. Zhong, H. Wu, L. Cheng, J. Lin, and Q. Huang, "A non-contact fall detection method for bathroom application based on mems infrared sensors," *Micromachines*, vol. 14, no. 1, p. 130, 2023.

[85] W. Min, H. Cui, H. Rao, Z. Li, and L. Yao, "Detection of human falls on furniture using scene analysis based on deep learning and activity characteristics," *IEEE Access*, vol. 6, pp. 9324–9335, 2018.

[86] X. Luo, T. Liu, J. Liu, X. Guo, and G. Wang, "Design and implementation of a distributed fall detection system based on wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, pp. 1–13, 2012.

[87] S. Gasparrini, E. Cippitelli, S. Spinsante, and E. Gambi, "A depth-based fall detection system using a kinect sensor," *Sensors*, vol. 14, no. 2, pp. 2756–2775, 2014.

[88] A. Naser, A. Lotfi, and J. Zhong, "Towards human distance estimation using a thermal sensor array," *Neural Computing and Applications*, vol. 35, no. 32, pp. 23 357–23 367, 2023.

[89] B. L. Moreland, R. Kakara, Y. K. Haddad, I. Shakya, and G. Bergen, "A descriptive analysis of location of older adult falls that resulted in emergency department visits in the united states," *American journal of lifestyle medicine*, vol. 15, no. 6, pp. 590–597, 2021.

[90] I. Nikolov, J. Liu, and T. Moeslund, "Imitating emergencies: Generating thermal surveillance fall data using low-cost human-like dolls," *Sensors*, vol. 22, no. 3, p. 825, 2022.

[91] Z. Tang, W. Ye, W.-C. Ma, and H. Zhao, "What happened 3 seconds ago? inferring the past with thermal imaging," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 111–17 120.

[92] C. Silver and T. Akilan, "A novel approach for fall detection using thermal imaging and a stacking ensemble of autoencoder and 3d-cnn models," in *2023 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, 2023, pp. 71–76.

[93] G. Baldewijns, G. Debard, G. Mertes, B. Vanrumste, and T. Croonenborghs, "Bridging the gap between real-life data and simulated data by providing a highly realistic fall dataset for evaluating camera-based fall detection algorithms," *Healthcare technology letters*, vol. 3, no. 1, pp. 6–11, 2016.

[94] U. Asif, S. Von Cavallar, J. Tang, and S. Harrer, "Sshfd: Single shot human fall detection with occluded joints resilience," in *ECAI 2020*. IOS Press, 2020, pp. 2656–2663.

[95] M. Gietzelt, J. Spehr, Y. Ehmen, S. Wegel, F. Feldwieser, M. Meis, M. Marschollek, K.-H. Wolf, E. Steinhagen-Thiessen, and M. Gövercin, "Gal@ home." *Zeitschrift fuer Gerontologie und Geriatrie*, vol. 45, no. 8, 2012.

[96] A. Sixsmith and N. Johnson, "A smart sensor to detect the falls of the elderly," *IEEE Pervasive computing*, vol. 3, no. 2, pp. 42–47, 2004.

[97] L. Badarch, M. Gochoo, G. Batnasan, F. Alnajjar, and T.-H. Tan, "Ultra-low resolution infrared sensor-based wireless sensor network for privacy-preserved recognition of daily activities of living," in *2021 IEEE 20th International Symposium on Network Computing and Applications (NCA)*, 2021, pp. 1–5.

[98] A. Rezaei, A. Mascheroni, M. C. Stevens, R. Argha, M. Papandrea, A. Puiatti, and N. H. Lovell, "Unobtrusive human fall detection system using mmwave radar and data driven methods," *IEEE Sensors Journal*, vol. 23, no. 7, pp. 7968–7976, 2023.

[99] M.-K. Yi, K. Han, and S. O. Hwang, "Fall detection of the elderly using denoising lstm-based convolutional variant autoencoder," *IEEE Sensors Journal*, 2024.

[100] A. Roy, R. Mukherjee, S. Moulik, and A. Chakrabarti, "Human fall prediction using ensemble learning technique," in *2022 IEEE International Conference on Consumer Electronics-Taiwan*. IEEE, 2022, pp. 545–546.

[101] S. Kordnoori, A. Sharifi, and H. Shah-Hosseini, "Human fall detection using neuro-fuzzy models based on ensemble learning," *Progress in Artificial Intelligence*, vol. 11, no. 3, pp. 219–232, 2022.

[102] G. Vavoulas, M. Pediaditis, E. G. Spanakis, and M. Tsiknakis, "The mobifall dataset: An initial evaluation of fall detection algorithms using smartphones," in *13th IEEE International Conference on BioInformatics and BioEngineering*. IEEE, 2013, pp. 1–4.

[103] G. L. Santos, P. T. Endo, K. H. d. C. Monteiro, E. d. S. Rocha, I. Silva, and T. Lynn, "Accelerometer-based human fall detection using convolutional neural networks," *Sensors*, vol. 19, no. 7, p. 1644, 2019.

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

[105] X. Yu, S. Park, D. Kim, E. Kim, J. Kim, W. Kim, Y. An, and S. Xiong, "A practical wearable fall detection system based on tiny convolutional neural networks," *Biomedical Signal Processing and Control*, vol. 86, p. 105325, 2023.

[106] X. Yu, J. Jang, and S. Xiong, "A large-scale open motion dataset (kfall) and benchmark algorithms for detecting pre-impact fall of the elderly using wearable inertial sensors," *Frontiers in Aging Neuroscience*, vol. 13, p. 692865, 2021.

[107] A. Sucerquia, J. D. López, and J. F. Vargas-Bonilla, "Sisfall: A fall and movement dataset," *Sensors*, vol. 17, no. 1, p. 198, 2017.

[108] T. R. Mauldin, M. E. Canby, V. Metsis, A. H. Ngu, and C. C. Rivera, "Smartfall: A smartwatch-based fall detection system using deep learning," *Sensors*, vol. 18, no. 10, p. 3363, 2018.

[109] D. Micucci, M. Mobilio, and P. Napoletano, "Unimib shar: A dataset for human activity recognition using acceleration data from smartphones," *Applied Sciences*, vol. 7, no. 10, p. 1101, 2017.

[110] C. Kittiyanpunya, P. Chomdee, A. Boonpoonga, and D. Torrungrueng, "Millimeter-wave radar-based elderly fall detection fed by one-dimensional point cloud and doppler," *IEEE Access*, 2023.

[111] C. Taramasco, T. Rodenas, F. Martinez, P. Fuentes, R. Munoz, R. Olivares, V. H. C. De Albuquerque, and J. Demongeot, "A novel monitoring system for fall detection in older people," *Ieee Access*, vol. 6, pp. 43 563–43 574, 2018.

[112] E. Auvinet, L. Reveret, A. St-Arnaud, J. Rousseau, and J. Meunier, "Fall detection using multiple cameras," in *2008 30th annual international conference of the ieee engineering in medicine and biology society*. IEEE, 2008, pp. 2554–2557.

[113] I. Charfi, J. Miteran, J. Dubois, M. Atri, and R. Tourki, "Optimised spatio-temporal descriptors for real-time fall detection: comparison of svm and adaboost based classification," *Journal of Electronic Imaging (JEI)*, vol. 22, no. 4, p. 17, 2013.

[114] K. Adhikari, H. Bouchachia, and H. Nait-Charif, "Activity recognition for indoor fall detection using convolutional neural network," in *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*. IEEE, 2017, pp. 81–84.

[115] S. Li and X. Song, "Future frame prediction network for human fall detection in surveillance videos," *IEEE Sensors Journal*, vol. 23, no. 13, pp. 14 460–14 470, 2023.

[116] J. Nogas, S. S. Khan, and A. Mihailidis, "Fall detection from thermal camera using convolutional lstm autoencoder," in *Proceedings of the 2nd workshop on aging, rehabilitation and independent assisted living, IJCAI workshop*, 2018.

[117] X. Ma, H. Wang, B. Xue, M. Zhou, B. Ji, and Y. Li, "Depth-based human fall detection via shape features and improved extreme learning machine," *IEEE journal of biomedical and health informatics*, vol. 18, no. 6, pp. 1915–1922, 2014.

[118] F. A. Elshwemy, R. Elbasiony, and M. T. Saidahmed, "A new approach for thermal vision based fall detection using residual autoencoder." *International Journal of Intelligent Engineering & Systems*, vol. 13, no. 2, 2020.

[119] Y. S. Chong and Y. H. Tay, "Abnormal event detection in videos using spatiotemporal autoencoder," in *Advances in Neural Networks-ISNN 2017: 14th International Symposium,*

ISNN 2017, Sapporo, Hakodate, and Muroran, Hokkaido, Japan, June 21–26, 2017, Proceedings, Part II 14. Springer, 2017, pp. 189–196.

[120] M. S. Seyfioğlu, A. M. Özbayoğlu, and S. Z. Gürbüz, "Deep convolutional autoencoder for radar-based classification of similar aided and unaided human activities," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 4, pp. 1709–1723, 2018.

[121] S. Lee, H. G. Kim, and Y. M. Ro, "Stan: Spatio-temporal adversarial networks for abnormal event detection," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 1323–1327.

[122] N. Hnoohom, A. Jitpattanakul, P. Inluergsri, P. Wongbudsri, and W. Ployput, "Multi-sensor-based fall detection and activity daily living classification by using ensemble learning," in *2018 International ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI-NCON)*. IEEE, 2018, pp. 111–115.

[123] Z. Mohammad, A. R. Anwary, M. F. Mridha, M. S. H. Shovon, and M. Vassallo, "An enhanced ensemble deep neural network approach for elderly fall detection system based on wearable sensors," *Sensors*, vol. 23, no. 10, p. 4774, 2023.

[124] Y. Ogawa and K. Naito, "Fall detection scheme based on temperature distribution with ir array sensor," in *2020 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2020, pp. 1–5.

[125] R. Liu and C. Vondrick, "Humans as light bulbs: 3d human reconstruction from thermal reflection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 531–12 542.

[126] M. Saleh, M. Abbas, and R. B. Le Jeannes, "Fallalld: An open dataset of human falls and activities of daily living for classical and deep learning applications," *IEEE Sensors Journal*, vol. 21, no. 2, pp. 1849–1858, 2020.

[127] S. Rastogi and J. Singh, "Performance enhancement of vision based fall detection using ensemble of machine learning model," *Cluster Computing*, vol. 26, no. 6, pp. 4119–4132, 2023.

[128] S. Shukralia, M. Bhatia, and P. Chakraborty, "Fall detection of elderly in ambient assisted smart living using cnn based ensemble approach," in *E-business technologies conference proceedings*, vol. 3, no. 1, 2023, pp. 134–139.

[129] A. Tahir, G. Morison, D. A. Skelton, and R. M. Gibson, "A novel functional link network stacking ensemble with fractal features for multichannel fall detection," *Cognitive Computation*, vol. 12, no. 5, pp. 1024–1042, 2020.

[130] N. El-Bendary, Q. Tan, F. C. Pivot, and A. Lam, "Fall detection and prevention for the elderly: A review of trends and challenges," *International Journal on Smart Sensing and Intelligent Systems*, vol. 6, no. 3, pp. 1230–1266, 2013.

[131] E. E. Stone and M. Skubic, "Fall detection in homes of older adults using the microsoft kinect," *IEEE journal of biomedical and health informatics*, vol. 19, no. 1, pp. 290–301, 2014.

[132] G. Debard, M. Mertens, M. Deschodt, E. Vlaeyen, E. Devriendt, E. Dejaeger, K. Milisen, J. Tournoy, T. Croonenborghs, T. Goedemé *et al.*, "Camera-based fall detection using real-world versus simulated data: How far are we from the solution?" *Journal of Ambient Intelligence and Smart Environments*, vol. 8, no. 2, pp. 149–168, 2016.

[133] S. Khan, "Classification and decision-theoretic framework for detecting and reporting unseen falls," Ph.D. dissertation, University of Waterloo, 2016.

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

[135] X. Peng and C. Schmid, "Multi-region two-stream r-cnn for action detection," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*. Springer, 2016, pp. 744–759.

[136] N. Lu, Y. Wu, L. Feng, and J. Song, "Deep learning for fall detection: Three-dimensional cnn combined with lstm on video kinematic data," *IEEE journal of biomedical and health informatics*, vol. 23, no. 1, pp. 314–323, 2018.

[137] C. Medrano, R. Igual, I. Plaza, and M. Castro, "Detecting falls as novelties in acceleration patterns acquired with smartphones," *PloS one*, vol. 9, no. 4, p. e94811, 2014.

[138] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?" in *ICML*, vol. 2, no. 3, 2021, p. 4.

Appendix

A Permission to Reprint

A.1 IEEE Permission to Reprint

In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of Lakehead University's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html and https://www.ieee.org/publications/rights/author-rights-responsibilities.html to learn how to obtain a License from RightsLink.

A.2 Elsevier Permission to Reprint

In reference to Elsevier copyrighted material which is used with permission in this thesis, Elsevier does not endorse any of Lakehead University's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing Elsevier copyrighted material for advertising or promotional purposes, or for creating new collective works for resale or redistribution, please go to https://www.elsevier.com/about/policies/copyright/permissions to learn how to obtain a License from RightsLink.

A.3 ICML Permission to Reprint

Portions of this thesis are reprinted from the author's paper submitted to the International Conference on Machine Learning (ICML). The author retains the copyright of this material. ICML holds a non-exclusive, perpetual license to distribute and publicly display the work if accepted. In this case, reprinting or republishing this material for advertising, promotional purposes, or creating new collective works for resale or redistribution requires permission. For more information, please visit https://icml.cc/FAQ/Copyright.

B Source Code

The source codes of this thesis are available on GitHub.

For more information about the author's publications, please refer to Google Scholar and LinkedIn profiles.

Google Scholar: Google Scholar.

LinkedIn: LinkedIn.

C Dataset Caching

C.1 Caching Performance Analysis

Table C.1: Average execution time, speedup, and dataset size for loading the dataset as cached images, images from disk, and NumPy arrays from disk

	Cache	Images	Numpy Arrays
Average Execution Time (s)	0.642	54.938	68.710
Speedup ($\times$)	94.986	1.000	0.814
Dataset Size (GB)	1.494	2.141	37.172

C.2 Script Explanation

The script systematically traverses the TF-66 directory, identifying all `.jpg` files within each directory (in this case, `train` and `validation`). Each image is then loaded using the `load_img()` method, resized to the target size (256×256), and converted to the specified color mode (grayscale). The processed images are stored in a cache dictionary, with the image paths as keys and the processed image arrays as values. The cache dictionary is then saved as a compressed `.npz` file using the `np.savez_compressed()` function. The script returns the cache dictionary, containing all the processed images. Upon completion, two cache files are generated: `train_cache.npz` and `val_cache.npz`. To load these cache files, simply invoke the `np.load` method. This script can be found on the associated github page, in a file named `Create_Cache.ipynb`.

C.3 Cache Sharing Considerations and Challenges

To address the need for creating a cache to efficiently work with the dataset, an attempt was made to cache the dataset on Google Drive in the base directory (`MyDrive`), and then share both the dataset and caches with another user. The goal was to enable other researchers to automatically utilize the cache in a Google Colab environment, offering a quick plug-and-play solution without requiring them to download the data or generate the cache themselves. The process involved sharing the cache files and the entire dataset folder from the sender to the recipient. In the recipient's Shared with me folder in Google Drive, they could right-click on the dataset folder, select Organize, and then Add shortcut, thereby adding it to their `MyDrive`. This setup ensured that both the sender and the recipient had access to the same cache and could use the dataset through an identical directory path. However, during testing, the recipient was unable to load images from the cache or train subsequent models using the cache, even though the file paths printed from the cache matched those found in their Google Drive. The reason for this issue was not explored in depth, but it is likely due to subtle differences in how Google Drive handles shared file paths or possibly due to differences in access permissions or file indexing. It is hoped that future work will address this limitation. In the meantime, and for researchers seeking an alternative to Google Colab, the script to create the cache can be found on the Github page with the file name "Create Cache.ipynb"

D Ideal Optical Flow Images

To ensure that the optical flow images generated by the Farneback method from the `cv2` library were optimized for use in machine learning models, multiple versions of the optical flow were generated and compared.

The template for the Farneback function call is as follows:

```
flow = cv2.calcOpticalFlowFarneback(prev_gray, next_gray, None,
                                    pyr_scale, levels, winsize,
                                    iterations, poly_n,
                                    poly_sigma, flags)
```

Where each parameter is defined as follows:

- `prev_gray` and `next_gray`: The two consecutive grayscale images between which the optical flow is calculated. These represent the previous and next frames in a sequence, respectively.
- `pyr_scale`: Pyramid scale factor, which specifies the image scale between pyramid layers. A higher value emphasizes smaller motions, making subtle movements more visible.
- `levels`: Number of pyramid layers. Increasing levels allows for capturing larger motions at varying scales, as each level reduces the image resolution to capture broader context.
- `winsize`: Window size for averaging over each pixel neighborhood. Smaller values capture finer details, while larger values are less sensitive to small motions but provide a smoother flow.
- `iterations`: Number of iterations the algorithm performs at each pyramid level. More iterations improve accuracy and allow the algorithm to refine smaller movements, but increase computation time.
- `poly_n`: Size of the pixel neighborhood for polynomial expansion. Smaller values increase sensitivity to fine changes, while larger values smooth the flow, reducing noise but potentially losing detail in subtle movements.
- `poly_sigma`: Standard deviation of the Gaussian that smooths the neighborhood. Higher values result in more smoothing and are less sensitive to small changes.

- `flags`: Optional flags to control various aspects of the optical flow algorithm, such as handling image borders. These can be used to fine-tune the algorithm based on specific requirements or constraints.

Five different versions of the Farneback method were used to extract the optical flow for the TF-66 dataset. In each version, the "sensitivity" of how much motion is captured in the resulting optical flow image is adjusted, by changing the different parameters in the `cv2.calcOpticalFlowFarneback` function.

The sensitivity levels and their corresponding parameters are found in Table D.2.

Table D.2: The five different sensitivity values and the corresponding parameter values in the `cv2.calcOpticalFlowFarneback`
Note: "G" - `cv2.OPTFLOW_FARNEBACK_GAUSSIAN`

Sensitivity Level	pyr_scale	levels	winsize	iterations	poly_n	poly_sigma	flags
High	0.8	5	15	5	5	1.1	0
Medium-High	0.7	4	19	4	7	1.2	0
Balanced	0.5	3	21	3	5	1.2	G
Low-Medium	0.4	3	25	3	7	1.3	G
Low	0.3	2	31	2	7	1.5	G

In order to evaluate which sensitivity level is most optical for the generated optical flow images, the generated optical flow images must be visually inspected. For each sensitivity level, two optical flow images are presented for comparison and evaluation. The first optical flow image represents motion. This optical flow image was generating by calculating the optical flow between frame 38 and 39 in fall video "01-Fall-04" from the TF-66 dataset. These frames represent an individual actively falling, and therefore should have a strong optical flow signature. These two images can be seen in the first row of images from Table D.3. The second optical flow image represents no motion, as it was generated by calculating the optical flow between frame 86 and 87 in fall video "01-Fall-04" from the TF-66 dataset, in which these two frames represent no motion, as the individual has already fallen and is laying in the prone position. These two images can be seen in the second row of images from Table D.3.

Table D.4 shows the resulting optical flow image generated for each sensitivity value. The right-most column shows demonstrates the optical flow generated for a person stationary on the ground, while the middle column shows the optical flow generated while someone is in the middle of a fall. In all sensitivity values except "Low", there was motion extracted when the person was laying on the ground, not moving. This is not wanted, and therefore the "Low" sensitivity value was selected. Examining the optical flow image where motion is supposed to be found, there is a large visual indication of motion in this spatial location where the fall occurred, which is desired for addition to the deep learning model, as optical flow is supposed to show when motion is happening between two consecutive frames. Therefore, the parameters in the `cv2.calcOpticalFlowFarneback` function were set to the "Low" sensitivity values found in Table D.2.

Table D.3: Comparison of consecutive frames illustrating motion and no motion in video "01-Fall-04" from the TF-66 Dataset

Note: The first row shows frames 38 and 39 demonstrating motion, while the second row shows frames 86 and 87 with no visible motion

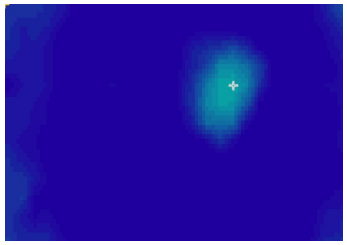
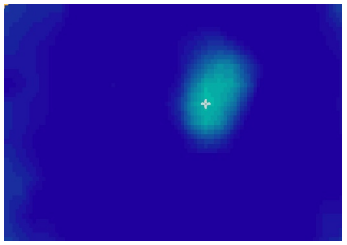
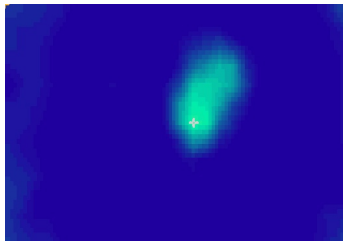
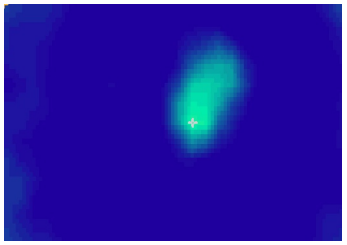
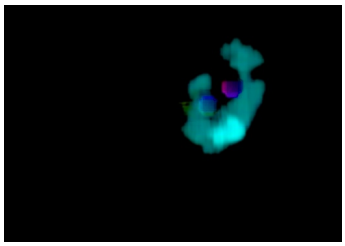
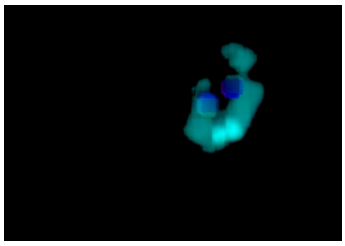
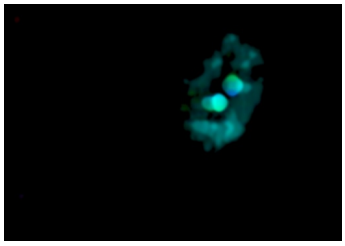
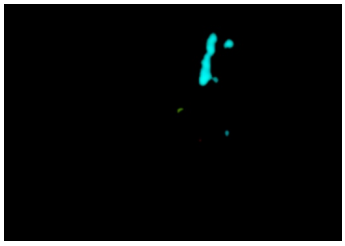
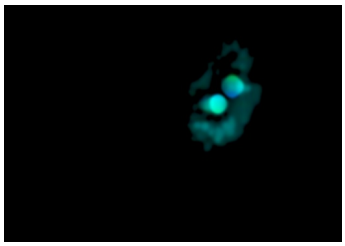
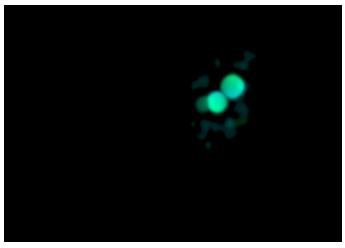
Description	Frame `X`	Frame `X+1`
Consecutive frames demonstrating motion		
Consecutive frames demonstrating no motion		

Table D.4: Comparison of optical flow outputs at varying sensitivity levels for frames 38-39 and 86-87 from video "01-Fall-04"

Sensitivity Level	Optical Flow Frames 38-39	Optical Flow Frames 86-87
High		
Medium-High		
Balanced		
Low-Medium		
Low		

E Future Development

For future development, we aim to deploy the optimized fall detection model on resource-limited platforms. In collaboration with Tushar Prasanna Swaminathan and Dr. Jitendra Kumar, I conducted a pilot study on optimizing deep learning models for edge devices like the NVIDIA Jetson Nano, with findings being prepared for publication. The study benchmarks DL architectures, including AlexNet, ResNet, and MobileNet-V2, alongside custom models like 3D CNNs and autoencoders, focusing on TensorRT optimization to improve inference speed and energy efficiency. As shown in Table E.5 and the accompanying chart E.1 which summarize the findings, MobileNet-V2 achieved the highest speedup (16.7×), demonstrating the effectiveness of these optimizations for real-time systems. Such advancements directly benefit fall detection applications by enabling low-latency processing on devices deployed in resource-constrained environments like elderly care facilities. These results provide insights into the trade-offs between model complexity, energy efficiency, and processing speed, emphasizing the importance of sustainable and scalable AI solutions for real-world edge applications.

Table E.5: Performance comparison of base models and their optimized versions on the NVIDIA Jetson Nano

Note: This table highlights the inference time improvements achieved through optimization, showing the speedup ratios, trainable parameters, and FLOPS. The experiments compare pre-optimization (Pre-Opt) and post-optimization (Post-Opt) results for both image classification and human action recognition models

Model	**Pre-Opt. Time (s)**	**Post-Opt. Time (s)**	**Pre/Post Time Ratio**	**Trainable Params**	**FLOPS**	**Inference Speed up**
Image Classification Models						
AlexNet	0.6638	0.1184	2.09	61,751,008	1,475,805,888	5.62×
VGG	2.5904	0.4285	6.05	143,667,240	19,590,954,654	6.05×
ResNet	1.0911	0.2233	4.88	25,557,032	3,883,453,952	4.88×
SqueezeNet	0.7966	0.1663	4.79	1,235,496	780,414,144	4.72×
DenseNet	1.0132	0.3682	2.16	7,978,856	2,845,996,288	2.72×
ShuffleNet V2	4.7121	0.3463	13.61	2,278,604	298,632,608	13.6×
MobileNet V2	5.0379	0.3003	16.77	3,504,872	300,356,480	16.7×
ResNet V2	2.0964	0.2600	8.06	25,028,904	3,866,082,816	8.07×
Human Action Recognition Models						
3D CNN	0.3431	0.0928	3.70	20,333,956	116,551,168	3.7×
AutoEncoder	0.7435	0.242	30.71	332,807	22,434,944	3.07×
DenseNet	3.1619	0.3718	8.50	109,386	27,681,924	8.5×

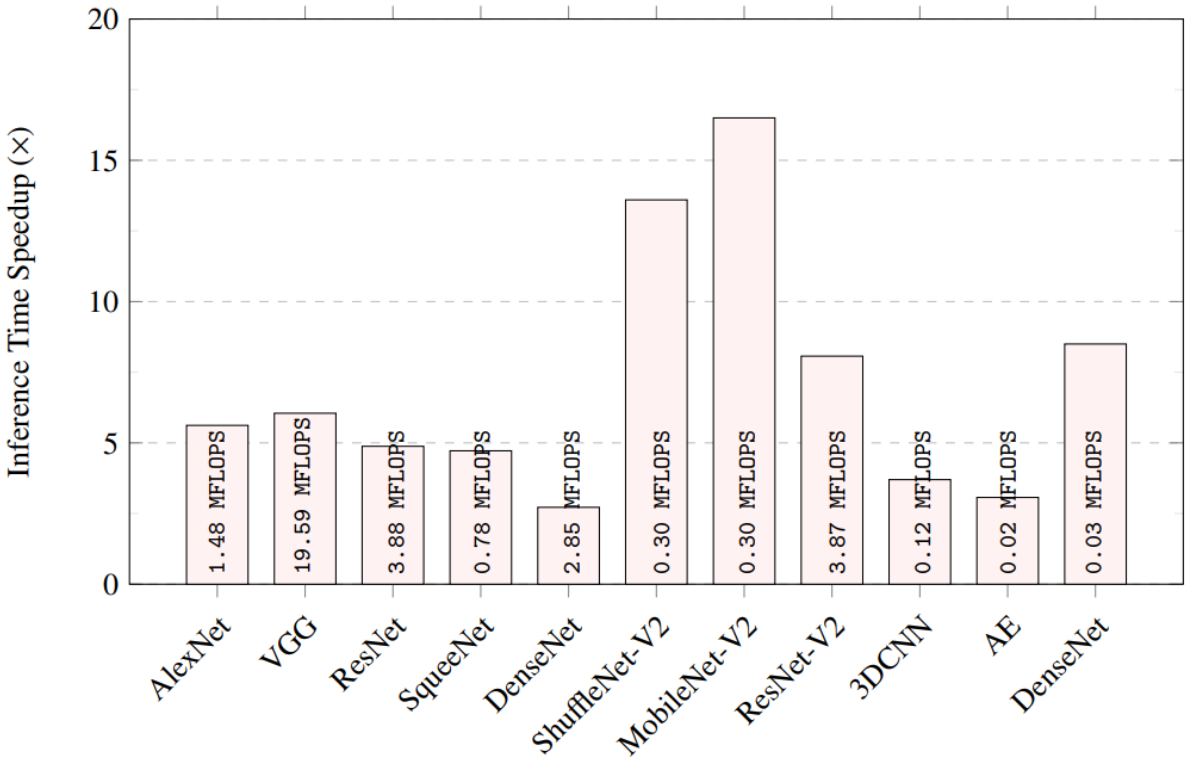


Figure E.1: Inference time speedup of optimized models compared to their non-optimized counterparts on NVIDIA Jetson Nano.