https://doi.org/ 10.33472/AFJBS.6.Si2.2024.632-645

African Journal of Biological Sciences

Journal homepage: http://www.afjbs.com

Research Paper Open Access

Real Time Fall Detection for Geriatric Risk Assessment using CNNs

RajBharath R[#1], Bhalamourale S[#2], Saaranathan D[#3], Suriya J[#4]

[#1,2,3,4] Department of Computer Science and Engineering,

[#1,2,3,4]Manakula Vinayagar Institute of Technology, Puducherry, India.

[1]rajbharathcse@mvit.edu.in

Article History

Volume 6,Issue Si2, 2024

Received:11 Mar 2024

Accepted : 12 Apr 2024

doi: 10.33472/AFJBS.6.Si2.2024.632-645

ABSTRACT

Falls are a significant threat to the geriatric population, and prompt intervention is crucial to ensure their safety. Existing fall detection methods often lack accuracy, especially in real-world settings with variable human motion and environmental factors. This paper proposes a novel approach to improve real-time fall detection for geriatric risk assessment. Our system utilizes an ADXL345 accelerometer to capture tri-axial acceleration data. An Arduino Uno serves as the primary microcontroller for signal collection, interpretation, and wireless data transmission via an ESP8266 Wi-Fi module to the ThingSpeak Internet of Things platform. This enables caregivers to remotely monitor geriatric individuals' movements in real-time. Additionally, data can be stored on ThingSpeak and subsequently fed into MATLAB for real-time fall detection using Convolutional Neural Networks (CNNs). Our approach leverages the feature extraction and pattern recognition capabilities of CNNs on sensor data from wearable devices to achieve accurate fall event identification. To enhance CNN robustness, we train them on a comprehensive dataset encompassing diverse fall scenarios and environmental conditions. Experimental results demonstrate significant improvements in fall detection accuracy compared to conventional techniques, highlighting the efficacy of CNNs in addressing real-world fall detection challenges. This study contributes to the advancement of fall detection technology, offering a more reliable solution for safeguarding the well-being of fall-prone individuals.

Keywords: Accelerometer sensor ADXL345, Arduino Uno, ESP8266 Wifi Module, ThingSpeak, MATLAB, CNNs.

1. INTRODUCTION

Unintentional or accidental falls are a big concern for people, particularly for the geriatric individuals, the disabled, and those with various injuries that make them weak. According to World Health Organization data, severe falls cause the deaths of over 5,96,000 persons year; the majority of these deaths occur in developing nations and involve adults 60 years of age or older. Age-related biochemical changes in the body make older persons weaker and more likely to faint or fall. Such falls may have extremely serious consequences, including lengthy hospital stays or, worse yet, early death. The impact of the fall determines how critical it is. If a person lies on the ground for an extended period of time following a fall, there is undoubtedly a problem with their health. Whether the wound is minor or severe, it is the duty of bystanders to assist the wounded. This is the situation in which a real-time fall detection system might be useful. Fall detection systems are designed to quickly identify potential falls and notify caregivers about them so that people can be saved as soon as possible.

It is quite sad that thousands of individuals pass away in this manner, unnoticed, and that lying down for an extended period of time can potentially result in death in cases of catastrophic falls. Therefore, creating effective fall detection systems is crucial in order to save lives.Real-time fall detection systems that precisely detects instances of falling in real time from triaxial accelerometer data by utilizing deep learning capabilities and promptly deliver notifications to enable medical aid to be rendered.By using a specially created CNN architecture that has been trained on a large dataset, ensures reliable performance in a variety of circumstances. By utilizing transfer learning strategies, the model exhibits excellent accuracy and broadening skills. The real-time capabilities of the system have the potential to improve proactive support and intervention, which in turn can help reduce the risks connected with falls and their accompanying consequences.

Thispaper provides a complete unique approachfor creating a sophisticated and trustworthy system for the automated, real-time detection of falls. There are several drawbacks to traditional fall detection techniques, such as rule-based algorithms, processing speed, accuracy, and adaptability. Despite its difficulties, computer vision-based fall detection has shown to be one of the most effective methods.Here, we suggest a real-time fall detection system that uses CNN-based strategy to get over these restrictions and present a more practical solution that can function flawlessly in a variety of natural settings.Overall, this research investigates the application of Convolutional Neural Networks (CNNs) for real-time fall detection, integrating with ThingSpeak for efficient data communication. This approach presents a significant advancement in fall detection technology, offering the potential to improve the safety and independent living of geriatric individuals. By enabling real-time fall recognition, this system can facilitate prompt intervention and medical assistance, potentially reducing fall-related injuries and promoting well-being for the geriatric population.

2. RELATED WORKS

Human Activity Recognition is a significant field of research in geriatric fall detection encompassing both computer vision and time series analysis approaches. There has been lot of research in recent times, focused on implementing fall detection technique in real time. We will highlight some of the most important efforts in fall detection systems in this overview of the literature.

This research conducted by B. R. Greene et al.'s [1] investigated the potential of combining clinical data and sensor information for fall risk assessment. It explores a statistical method for fall risk assessment using standard clinical factor form a dataset of 748 individuals. Additionally, they introduce an algorithm that leverages data from inertial sensors worn by participants, along with results from the Timed Up and Go test, a common clinical assessment of mobility. However, further research is needed to address the challenge of generalizability across diverse populations.

Bharathiraja et al.'s research paper [2] demonstrates the feasibility of using thermal sensors for real-time fall detection. Their system utilizes an AMG8833 thermal sensor to detect the changes in heat patterns associated within a specific area. By monitoring rapid changes in thermal signatures, the system can identify potential falls and trigger an alarm or notification. This approach holds promise for applications in assisted living facilities or elderly care homes where privacy and ease of use are crucial considerations.

This study by J.G. et al. proposes a vision-based fall detection system that utilizes Convolutional Neural Networks (CNNs) for real-time applications [3]. Their approach incorporates enhanced optical flow analysis to capture motion information and leverages rank pooling for efficient temporal data processing. This method enables the system to achieve high accuracy in fall detection even under dynamic lighting conditions, making it suitable for various real-world scenarios.

Rajalaxmi et al.'s research demonstrates a CNN-based approach for fall detection [4] that eliminates the need for wearable sensors, potentially improving user comfort and compliance. Their system utilizes optical flow images, which represent the motion between consecutive video frames, as input to the CNN. This approach addresses the challenge of limited fall detection datasets by employing transfer learning techniques. By leveraging pre-trained models on a different task, the CNN can learn essential features that generalize well to fall detection.

Sadreazami et al. explore the application of deep Convolutional neural network for fall detection using data from a StandOff Radar (STAR) sensor [5]. Unlike traditional vision sensors, STAR offers the advantage of working in low-light conditions and being less susceptible to occlusions. Their work proposes a multi-level feature learning approach directly from the radar time series data, enabling the CNN to effectively extract relevant features for fall detection. This research highlights the potential of alternative sensing modalities beyond cameras for real-time fall detection systems.

Saleh and Jeannès present a practical approach to fall detection using machine learning with wearable sensors [6]. Their focus on affordability and real-world implementation resonates with the potential applications of our CNN-based system where the real time fall detection utilizes the wearable devices for the transmission of data in real-time. While not utilizing CNNs, their work highlights the importance of considering cost-effectiveness and user comfort in designing fall detection solutions.

This research conducted by Zhang et al. delves into fall detection within video data [7]. They propose a novel descriptor that captures two key aspects i.e. Trajectory Information, a descriptor that incorporates information about the movement path of an individual within the video frame. By analyzing the trajectory, the system can identify sudden changes in motion patterns that might indicate a fall event and Reduced Temporal Redundancy, as videos contain a large amount of redundant information across consecutive frames. Their approach utilizes cluster pooling to efficiently reduce this redundancy. By grouping similar frames together, the system focuses on capturing the essential changes in motion over time, improving processing efficiency.

While not directly employing CNNs, the work by Zerrouki et al. [8] offers valuable insights by explores human action classification using the adaptive boosting algorithm, focusing on recognizing falls within video data. This study emphasizes the importance of efficient feature extraction for fall detection tasks. By extracting informative features from the video frames, the system can effectively distinguish between fall events and other human actions. This aligns with the role of CNNs in our proposed work, where feature extraction through convolutional layers plays a crucial role in accurate fall detection.

Deep learning application for detecting the fall conducted by Min et al. [9] explores the challenge of fall detection in complex environments, particularly falls involving furniture. Their research utilizes deep learning for scene analysis, incorporating contextual information alongside video data. Their approach demonstrates the potential for improving fall detection accuracy by considering the surrounding environment and concept of incorporating additional contextual information could enhance the robustness of the system.

In their study, Kim et al. introduce practical approach for three-dimensional fall detection that leverages depth estimation techniques [10]. It concentrates on addressing the challenge of fall detection in outdoor environments using a single camera to capture video data and employs algorithms to estimate the depth information for each pixel in the frame. This creates a 3D representation of the scene, allowing the system to distinguish between a person standing, sitting, or lying on the ground, facilitating more accurate fall detection compared to solely relying on 2D video data. Additionally, the use of depth information eliminates the need for background subtraction techniques, which can be sensitive to environmental changes. This research demonstrates the potential of 3D vision for robust fall detection in outdoor environments.

Sehairi et al. present a fall detection system that combines motion analysis with multiple shape features extracted from video data [11]. The system analyzes the movement patterns of the individual in the video frame. This includes tracking changes in position, velocity, and acceleration. By identifying sudden or erratic movements, the system can detect potential fall events. In addition to motion analysis, the system extracts various shape features from the video data. This might include the aspect ratio, centroid location, and size of the person's body. By analyzing these features in conjunction with motion information, the system can gain a more comprehensive understanding of the person's posture and activity, which reduces the frequency of false alarms.

3. EXISTING WORK

A. Overview

A fall detection system can be categorized using a variety of factors. Certain classifications are made depending on the sensors—such as ambient light sensors, pressure sensors, cameras, etc.—that are used to capture fall data. Some rely on the numerous phases that follow the fall, while others are based on the fall's impact. In order to identify fall events, the majority of systems have employed camera-based (visual) and wearable sensor-based (non-visual) approaches. The fundamental components of all fall detection systems are the same: sensors must first record fall data, and then there must be an effective distinction made between falling and regular activities like walking, sitting, and standing.

B. Non-vision based fall detection

A non-vision strategy makes use of wearables with built-in sensors, such as microphones, accelerometers, gyroscopes, heart rate monitors, or magnetometers, to detect falls. Gyroscopes and accelerometers are two common types of sensors. These sensors monitor the user's posture and enable the collection of essential data from their motions and activities. Accelerometers are essential sensors in fall detection systems that detects irregular motion patterns associated with falling when there is an abrupt deviation from this baseline, both in terms of intensity and direction. Another effective sensor for fall detection is the gyroscope. Gyroscopes are primarily used in fall detection systems to differentiate between intentional movement and unintentional falls. When combined, they could improve accuracy in activities like fall detection. To detect falls, these sensors are attached to the subject's waist, thigh, wrist, shoes, etc.Threshold-based algorithms are used in a non-vision approach to analyze wearable sensor data. Fall events in threshold-based algorithms can be separated into three stages: pre-fall, impact, and post-fall. If one or more characteristics of the sensor data surpass a predetermined threshold, a fall can be identified.

Significantly, when applied to real-world data, algorithms with higher fall detection rates on the simulated falls also typically had higher false alarm rates. These kinds of mistakes are to be expected since, in reality, a lot more events can match these thresholds because of high acceleration, but these threshold-based algorithms are only categorizing events with a maximum magnitude as falls. These kinds of mistakes are particularly common in threshold-based algorithms.On the other hand, geriatric individuals might not be that interesting to constantly be observed or monitored. In addition, it compels the wearer of the sensors to actively co-operate, which can be difficult and possibly painful. In addition to the risk of long-term skin contact, people may forget to wear these sensing devices. Furthermore, these sensors are unreliable since they frequently generate a high number of false alarms. As a result, less invasive and demanding vision-based methods have been suggested.

C. **Vision based fall detection**

Fall detection can be accomplished by using visual sensors, such as cameras,instead of continuously attaching sensors to the body. In order to identify possible falls, vision-based fall detection techniques use visual sensors to track and analyze human movement. Video is analyzed by these systems to extract relevant information within the environment, and posture of the body. It is possible to detect falls without constantly attaching sensors to the body by employing sight sensors, like cameras. Utilizing video analysis, these systems derive pertinent attributes like motion patterns, spatial relationships in the environment, and body posture. Many other sorts of features exist, such as aspect ratio, head position, orienting angle, shape, center of mass, and distance from the floor to the centroid of the human body. Computer vision techniques, including neural networks and classifiers, are then used to identify unique features linked to a fall. To some extent, the issue of ambience-based and wearable sensor-based system is resolved by vision-based fall detection systems. Depth sensor and pressure sensor plays a major role in vision-based fall detection approach.

Depth sensors are used in fall detection because they can identify a person's height at which they fall, their point of contact with the floor, and their following motions in fall detection scenarios. This information is useful in situations where traditional cameras alone might struggle, such as low light or situations with complex backgrounds.Pressure sensors are used to detect the sudden rise in pressure during a fall occurrence by placing it strategically on the floor or on furniture surfaces. This data adds another layer of information that fall detection algorithms might use to improve their capacity to differentiate between normal activities and potentially harmful incident.On the other hand, challenges include problems with lighting conditions, privacy, and obstructions that could block the camera's field of vision. Sustained advancement of computer vision, machine learning, and sensor technologies enhances the precision and dependability of vision-based fall detection

systems, making them feasible options in different environments like residences, healthcare organizations, or assisted living areas.

4. PROPOSED SYSTEM

A. Overview

The primary goal of the system's design is to create a human fall detector that can use the CNN model to reduce the many steps of typical image processing systems and achieve state-of-the-art accuracy.As was previously mentioned, there are a number of issues with fall detection systems based on wearable sensors and ambiance sensors that make them less useful for everyday use. The accuracy of fall detection is enhanced by the CNN based fall detection design as it is proved that CNN is one of the best image classifiers and is being widely used in image and video processing. In this approach, the system that was built may eliminate false alarms or lower the quantity of false positives, and it could be applied to create a fall detection system that is more dependable in real-world settings.

B. Dataset Description

This dataset was created for training and evaluating a Convolutional Neural Network (CNN) model for real-time fall detection for geriatric individuals. It contains data points, each representing a single instance of sensor data collected from a triaxial accelerometer.

TABLE I. DATASET DESCRIPTION

CREATED_AT	This column stores the date and time (in UTC) when the data point was collected.
ENTRY_ID	This column assigns a unique identifier to each data point.
FIELD1	This column represents the triaxial acceleration values measured along the x-axis
FIELD2	This columnrepresents the triaxial acceleration values measured along the y-axis
FIELD3	This column represents the triaxial acceleration values measured along the z-axis
LABEL	This column indicates whether the corresponding sensor data represents a fall event(labelled 1) or not (labelled 0).

C. Architecture of Neural Network

1) Feature Input Layer

Feature input layer acts as the entry point for the data. It expects three features or channels which could represent acceleration values in three axes (x axis, y axis and z axis) from your sensor. The “z-score” normalization ensures all features have a zero mean and unit standard deviation. This helps the network learn more effectively by placing all features on an equal scale. The z-score normalization can be calculated as:

$$Normalized\_feature = (original\_feature - mean(all\_features)) / standard\_deviation(all\_features)$$

2) Fully Connected Layer

There are three hidden fully connected layer of which the first hidden fully connected layer takes the normalized three-dimensional data and performs a linear transformation using 40 weights and baises. This creates 40 new features or activations that capture more complex relationships between the original features. These features could represent combinations of acceleration values in different axes, potentially identifying patterns indicative of movement or a fall.Then, there is the second hidden fully connected layer with 10 neurons that takes the output from the ReLU layer and performs another linear transformation with ten weights and biases. This creates 10 new features that represent even more complex combinations of the original features. Atlast, there is final fully connected layer with 2 neurons. Since, there are 2 classes i.e. fall and not fall, this layer transforms the data into a two-dimensional vector. Each element represents the probability of the input belonging to a

specific class. This layer performs a linear transformation using weights and biases. Let X be the input vector, W be the weight matrix and b be the bias vector. The output, Y is calculated as:

$$Y = W * X + b$$

3) Batch Normalization Layer

Batch Normalization layer applies batch normalization. During training, it normalizes the activations of the previous layer across a mini-batch of data. This helps improve training speed and stability by reducing the internal covariate shift. The second batch normalization layer normalizes the activations of the second fully connected layer as like the previous layer. It normalizes the activations using parameters like batch mean and standard deviation, which is given as follows:

$$Normalized\_x\_i = (x\_i - \mu) / sqrt (\sigma^2 + \varepsilon)$$

where μ is the batch mean and σ^2 is the variance, can be calculated as

$$\mu = (1 / m) * \Sigma(x\_i) \quad \text{for all i in the mini-batch}$$

$$\sigma^2 = (1 / m) * \Sigma[(x\_i - \mu)^2] \quad \text{for all i in the mini-batch}$$

where, m is the number of samples in the mini batch and x_i = activation value for a specific feature in sample i

4) ReLU Layer

This layer applies the ReLU (Rectified Linear Unit) activation function. ReLU sets negative values to zero and keeps positive values unchanged. This introduces non-linearity, allowing the network to learn more complex relationships between features. ReLU might help focus on acceleration values that could be indicative of sudden movements during a fall. The ReLU activation function applies a threshold to the outputs from the previous layer (Y) and it is given by,

$$Output = max (0, Y)$$

5) Softmax Layer

The SoftMax layer converts the outputs from the last fully -connected layer into probability distributions. The sum of the outputs will be 1, with the first value representing the probability of being a fall and the second value representing a probability of not being a fall. It takes the output vector (Y) from the last fully-connected layer and converts it into probability distribution. The SoftMax function applied to each element yi in Y as:

$$Softmax(yi) = exp(yi) / sum(exp(yj) \text{ for all j in Y})$$

6) Classification Layer

The classification layer performs classification based on the softmax outputs. It likely assigns the class with the highest probability as the predicted class fall or not fall.

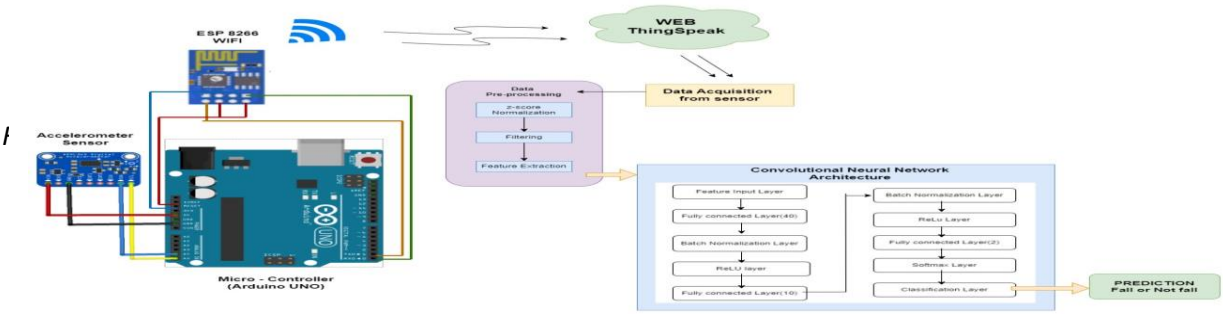

Figure 1. System Architecture Diagram

D. System Architecture

The proposed system entails the development of a robust fall detection framework utilizing MATLAB, Arduino microcontroller, accelerometer sensor, Wi-Fi connectivity, andConvolutional Neural Network, a deep learning model.The main goal of this system is to monitor and store data via the Internet of Things Platform.The system makes use of a triaxial ADXL345 accelerometer, which measures an elderly person's acceleration continuously along three perpendicular axes: the X, Y, and Z. In contrast to regular activities like standing or walking, the geriatricindividual's body undergoes a quick and dramatic change in acceleration during the fall. These variations in acceleration are recorded by the ADXL345 on all three axes, giving useful information for fall detection. The Nyquist-Shannon sampling theorem states that the sensor may supply the microcontroller unit (Arduino UNO) in the fall detection system with its raw acceleration data at a sampling rate of 40 Hz, which is adequate for fall detection. Because less power is required for acquisition when using a low sampling frequency. This range provides an excellent compromise between the efficiency of data processing and data capture, and it is formatted in 16 bits. After reading the sensor data, the Arduino Uno preprocesses it by filtering out undesired fluctuations and noises, and normalizing the data, which means scaling numbers between -1 and 1 to reflect gravitational force. The Arduino Uno and the internet are connected via the ESP8266. It enables the Arduino Uno to establish a wireless network connection and sends real-time, normalized data to ThingSpeak. This platform has no usage limits and updates data every 15 seconds. It allows for analysis and visualization by storing the data in its channels. ThingSpeak displays the data in real time as graphs along the x, y, and z axes, enabling us to see patterns of acceleration during falls and other activities.The basis for building our own fall detection dataset is provided by these graphs. Afterwards, the CNN model is trained using this ThingSpeak dataset to identify fall patterns based on the acceleration data along three axes. Using the ThingSpeak dataset, a convolutional neural network model is trained to detect falls. The model is trained and improved to reliably categorize fall and non-fall events based on the retrieved data by utilizing MATLAB’s extensive deep learning toolbox. When angeriatricindividuals engages in activities, the Arduino gathers fresh acceleration data in real-world circumstances. Through the ESP8266 wifi module, the data is processed and delivered to the trained CNN model. The CNN model generates a conclusion that is either a fall or not by comparing the real-time data with the patterns it has learned from the dataset. The CNN model can be modified for better fall detection accuracy by utilizing the real-time data that has been gathered. Thereby, the system provides a comprehensive fall detection by combining the analytical power of MATLAB with hardware interfacing capabilities to produce a stable and effective fall detection framework.

5. METHODOLOGY

A) Data Acquisition

The data acquisition module forms the foundation of this fall detection system. It is responsible for capturing the raw acceleration data that will be used to identify falls.

Sensor Initialization: The Arduino Uno first establishes communication with the triaxial accelerometer, the ADXL345. This involves initializing the sensor and configuring its settings. This ensures the sensor operates correctly and transmits data in the expected format.

Continuous Acceleration Measurement: Once communication is established, the Arduino starts reading data from the accelerometer at regular intervals. The ADXL345 measures acceleration along three perpendicular axes: X (front-to-back), Y (side-to-side), and Z (up-down). With each reading, the Arduino receives three values representing the acceleration forces acting on each axis. The core function of the data acquisition module is to prepare the raw sensor data for transmission to the next stage. This might involve basic formatting or packaging of the data into a structure suitable for wireless transmission.The Arduino reads the sensor data at a sampling rate of 40 Hz, which is adequate for fall detection. Because less power is required for acquisition when a low sampling frequency is used. A higher sampling rate provides more detailed data but requires more processing power and storage. A balance needs to be struck between capturing enough detail and keeping the system efficient.

B) Data Preprocessing and Transmission

The data acquisition module is responsible for collecting the raw acceleration data from the triaxial accelerometer. However, this raw data might contain noise, fluctuations, and may not be directly usable for fall detection. So, the data preprocessing module comes into play. The data preprocessing module takes the raw acceleration data from the data acquisition module and performs several key tasks such as:

Noise Filtering: Real-world sensor readings are often accompanied by electrical noise or environmental disturbances. The preprocessing module applies filters to remove this noise, ensuring the data accurately reflects the user's movements.

Data Normalization: The raw acceleration data might be measured in voltage units specific to the sensor. The preprocessing module normalizes the data between a standard range, typically -1 to 1. A value of 1 signifies an acceleration equal to gravity (1g), while -1 indicates an acceleration in the opposite direction. Values between -1 and 1 represent varying degrees of acceleration force. This simplifies further processing and analysis by representing the data in terms of the force of gravity acting on each axis. This processed data becomes the foundation for training the fall detection model and ultimately identifying fall events in real-time. The module might also format the data into a specific structure or protocol suitable for transmission to the next stage i.e. data transmission

Data Transmission: After normalization, the Arduino prepares a string containing the normalized acceleration data for each axis. It utilizes the ESP8266 Wi-Fi module to achieve wireless communication with ThingSpeak. The data transmission process involves establishing a connection with ThingSpeak, formatting the data into a transmittable format i.e. string and sending the data over the internet connection. This module ensures the pre-processed data reaches the ThingSpeak platform for further analysis and storage. Upon receiving the data, ThingSpeak stores it in its channels and displays it in real time as graphs along the X, Y and Z axes. This allows for initial visualization and analysis of the acceleration pattern.

C) Feature Extraction

The feature extraction module aims to transform raw acceleration data from all three axes into a set of meaningful features that are most informative for the task of fall detection. It might employ various techniques to extract relevant features from the raw data.

Peak Acceleration: This feature identifies the maximum value of acceleration experienced in each axis during a specific time window. High peak acceleration might be indicative of falls.

Signal Variance: This feature measures the variation of the acceleration signal over time. Sudden changes in the variance could be associated with falls.

Zero-Crossing Rate: This feature counts the number of times the acceleration signal crosses the zero line within a time window. Rapid zero-crossing rates might suggest falls.

Moreover, not all extracted features might be equally important for fall detection. Feature selection techniques might be employed to identify the most relevant features that contribute the most to the model's performance. This can improve model efficiency and reduce computational burden. Finally, it generates a set of numerical features representing the extracted characteristics from the original sensor data and might combining the extracted features to create even more informative representations. This feature vector is then used as input for the CNN model during training and prediction phases.

D) Model Training

This module utilizes a Convolutional Neural Network to learn how to identify falls based on the extracted features from the normalized acceleration data. Following are the crucial steps involved in model training:

Data Selection: This stage involves gathering a large dataset of labelled fall and non-fall events by extracts the relevant columns from the loaded table of the ThingSpeak acceleration data.These likely represent the pre-processed acceleration data and a corresponding label i.e. fall or not fall from the original data and converts the table data into a numerical array.

Training and Testing Data Split: It defines the parameters like the number of samples per class, assuming two classes i.e. fall or not fall and the number of features. It then extracts the features and labels from the pre-processed data. It generates a random permutation of indices for all data points. This ensures the model is trained and tested on a representative subset of data to avoid overfitting. Finally, it splits the data into training and testing sets using the generated random indices. This allows the model to learn from the training data and evaluated on unseen data from the testing set.

Model Architecture: The architecture of the CNN model determines how the model processes information and learns to classify falls, like a multi-layered processor that analyzes sensor data. The input layer receives pre-extracted features, like peak acceleration, and the number of neurons in this layer matches the number of features. Fully connected layers, with 40 and 10 neurons each, perform complex calculations to unearth hidden relationships between these features and falls. Batch normalization layers inserted after each fully connected layer enhance training speed and stability. ReLU layers with their ReLU activation function introduce non-linearity, allowing the model to learn intricate patterns in the data. Finally, the output layer with two neurons and a softmax function acts as the decision maker, assigning probabilities to the "fall" or "not fall" classification based on the processed data.

E) Real time Processing

The real-time processing module continuously monitors for falls using a pre-trained CNN as it retrieves data from ThingSpeak at regular intervals. This data likely originates from a wearable sensor worn by the user. The model was previously trained on a large dataset of labeled fall and non-fall events. During training, the model learned to identify patterns in the extracted features that are indicative of falls. The retrieved data is fed into the loaded CNN model as it analyzes the features in real time and predicts whether a fall has occurred or not. It outputs a label, likely "fall" or "not fall", based on its classification. The system retrieves new data, performs predictions and displays results at regular intervals, enabling real-time fall detection.Moreover, implement a monitoring system is needed to continuously assess the model's performance in real-time.If performance degrades or if the system encounters new patterns, trigger adaptations such asmodelretrainingorupdates toensureongoingeffectiveness.

6. RESULTS

This section presents the evaluation results of various machine learning algorithms tested on the real-time fall detection dataset in comparison with our proposed algorithm. The dataset comprised of number of data points containing triaxial acceleration measurements (X, Y, Z axes) and corresponding fall labels (fall: 1, not fall: 0). The outcomes show the advantages and disadvantages of each strategy, thereby provides whether CNN is a better option than other conventional machine learning methods.

TABLE II. PERFORMANCE OF VARIOUS ALGORITHM OVER FALL DETECTION

Model	Accuracy	Precision	Recall	F1-score
Logistic Regression	89	85	86	85.49
Gaussian Naive Bayes	95	92	91	91.49
Decision Tree	87	86	85	85.49
K Nearest Neighbors	94	92	90	90.99
CNN	98	94	95	94.5

The Table II shows that, in the evaluation of various algorithm on thingspeak fall detection dataset, three algorithms emerged as the top performers.CNN, K Nearest Neighbors and Gaussian Naïve Bayes achieves a precision score of 94%, 92% and 92% respectively. When it comes to recall, the CNN takes driver seat with a recall score of 95%, followed by Gaussian Naïve Baiyes at 91%. Therefore, the Convolutional Neural Network (CNN) achieved the highest performance among all the tested algorithms on the fall detection dataset. The CNN achieved an accuracy of 98%, precision of 94%, recall of 95%, and F1-score of 94.5% which is better than the other fall detection algorithms. This indicates that the CNN model was able to correctly classify most of the fall and non-fall instances in the dataset, with a good balance between precision and recall.

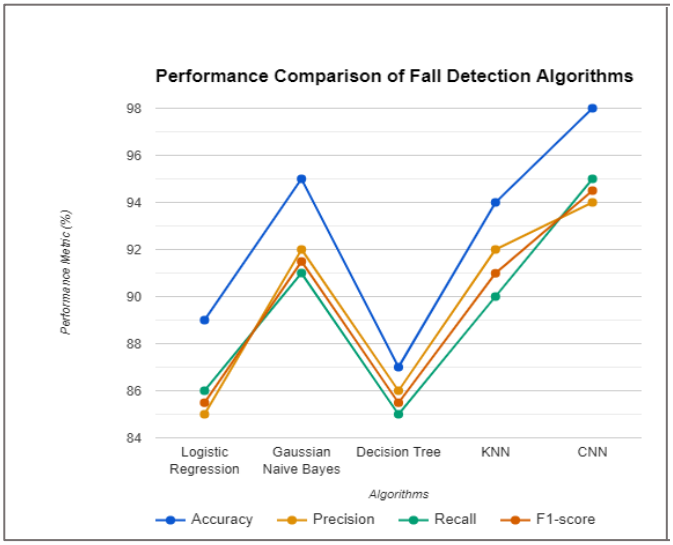

Figure 2. Performance metrics of various algorithms

The following is the ThingSpeak graph showing the acceleration data along the three axes:

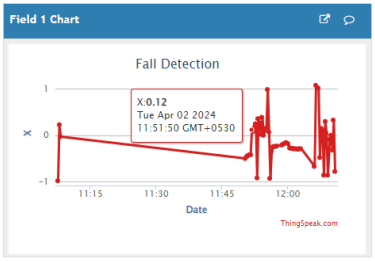

Figure 3: Normalized value graph for x axis

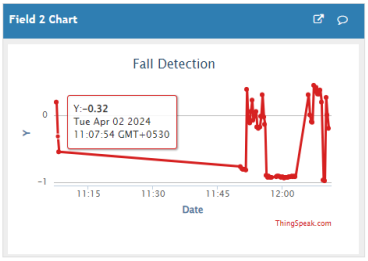

Figure 4: Normalized value graph for y axis

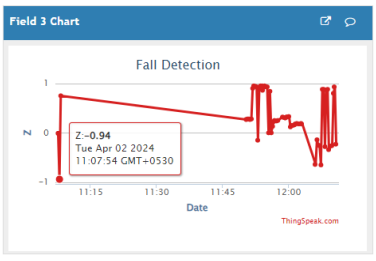

Figure 5: Normalized value graph for z axis

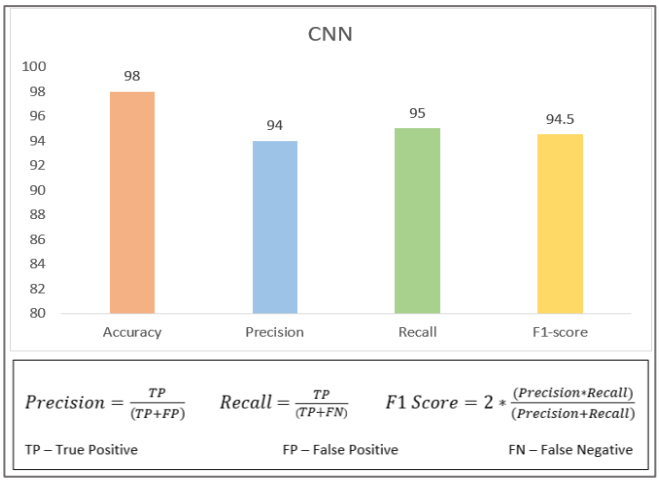

Figure 6. Performance metrics of Proposed Algorithm

Logistic Regression and Decision tree exhibited the lowest performance metrics across all categories. This suggests that these algorithms might not be well-suited for capturing the complex relationships within the falldetection data.The results demonstrate the effectiveness of Convolutional Neural Networks for real-time fall detection using triaxial accelerometer data. The CNN's superior performance highlights its ability to learn intricate patterns from the acceleration measurements, leading to accurate fall classification.

Date and Time (yy:mm:dd) (hh:mm:ss)	Fall Detection(Fall or Non-Fall)	X-axis Peak Acceleration (g)	Y-axis PeakAcceleration (g)	Z-axis Peak Acceleration (g)
2024-04-01 17:32:28	Non-Fall	0.91	0.32	0.36
2024-04-01 17:33:05	Fall	-0.75	-0.20	0.49
2024-04-01 17:33:56	Fall	-0.26	0.37	-0.87
2024-04-01 17:34:34	Non-Fall	0.25	0.05	0.93
2024-04-01 17:35:09	Non-Fall	0.39	-0.19	0.86
2024-04-01 17:35:47	Non-Fall	0	0.49	0.67
2024-04-02 06:46:23	Fall	0.23	-0.32	-0.94
2024-04-02 06:51:12	Non-Fall	0.55	-0.24	0.19
2024-04-02 06:55:46	Fall	-0.50	-0.77	0.27
2024-04-02 06:57:14	Fall	0.35	-0.69	0.13
2024-04-02 07:01:29	Fall	-0.25	-0.93	0.16
2024-04-02 07:02:07	Fall	-0.93	-0.69	0

TABLE III.REAL-TIME THINGSPEAK DATA ANALYSIS FOR FALL DETECTION

The Table III summarizes the ongoing monitoring results of the real-time fall detection system. The data was collected from a triaxial accelerometer worn by angeriatric individual. Each row represents a single instance captured during continuous monitoring where the Convolutional Neural Network (CNN) model analyzed the sensor data and classified it as a fall or non-fall event. By continuously collecting and analyzing the data, we can gain insights into the system's responsiveness to various movements and its ability to distinguish falls from daily activities. This information can be helpful in evaluating the overall effectiveness of the real-time fall detection system and potentially refining the CNN model for improved accuracy in the future.

7. DISCUSSION

This paper proposes a real-time fall detection system for geriatric individuals using an Arduino Uno microcontroller, a triaxial accelerometer, and a Convolutional Neural Network (CNN) model. The system offers several advantages over traditional fall detection methods, which can improve the safety and well-being of geriatricindividuals.

Enhanced Accuracy and Fall Pattern Recognition:

- Machine Learning with CNN: The system utilizes a CNN model trained on a fall detection dataset. This allows it to learn and recognize complex fall patterns in the acceleration data, potentially achieving higher accuracy compared to traditional methods that rely on simple thresholding or rule-based algorithms. These traditional methods may struggle to differentiate between falls and daily activities.
- Real-Time Data Analysis: Unlike some existing systems that rely on periodic data analysis, the proposed system continuously analyzes acceleration data in real-time. This enables faster detection of potential falls, allowing for quicker intervention and potentially improving health outcomes.

Improved User Comfort and Wearability:

- Non-invasive Sensor: The triaxial accelerometer is a small, lightweight sensor that can be comfortably worn by the person. This eliminates the discomfort or limitations associated with some traditional methods like wearable pressure mats or cameras.
- Wireless Communication: The ESP8266 Wi-Fi module enables wireless data transmission from the Arduino Uno to ThingSpeak. This eliminates the need for cumbersome wires, improving user comfort and mobility, especially for active geriatric individuals.

Remote Monitoring and Alerting:

- Cloud-based Data Storage (ThingSpeak): The system utilizes ThingSpeak for data storage and visualization. This allows caregivers or healthcare professionals to remotely monitor the geriatric individual's activity data and receive alerts in case of a suspected fall. This remote monitoring capability can provide peace of mind for caregivers and a sense of security for elderly users living alone.
- Scalability and Adaptability: The system can be potentially scaled to monitor multiple geriatric individuals by adding additional sensor units and integrating them with the cloud platform. Additionally, the CNN model can be continuously improved by incorporating new fall data into its training process, allowing for adaptation and improvement over time.

Cost-Effectiveness:

The system utilizes relatively inexpensive components like the Arduino Uno and triaxial accelerometer, making it a more accessible solution compared to some advanced fall detection systems. This cost-effectiveness can be particularly relevant for home-based elderly care or in situations where resources might be limited.

Data-driven Approach:

The CNN model provides valuable insights into fall patterns, potentially aiding in fall prevention strategies and risk assessments. By analyzing historical data, caregivers or healthcare professionals might be able to identify activities or situations that put the geriatricindividuals at a higher risk of falls.

CONCLUSION

Convolutional Neural Networks (CNNs) have emerged as a promising technique to enhance real-world fall detection accuracy for geriatric risk assessment. Our study demonstrates that CNNs effectively analyze data from the ThingSpeak platform to achieve precise fall event identification. This capability addresses critical safety concerns, particularly for the geriatric population.The proposed solution leverages the strengths of CNNs in feature extraction and pattern recognition, leading to significant improvements over conventional fall detection methods. The experimental results validate the efficacy of CNNs in addressing challenges associated with real-world fall detection scenarios. The significant accuracy gains achieved by the CNN models can be attributed to their robustness and generalization capabilities fostered by training on a comprehensive dataset. These findings highlight the transformative potential of CNNs in revolutionizing fall detection technology, ultimately contributing to safer living environments for fall-prone individuals.Future research and development efforts will focus on further refining CNN-based fall detection systems through class expansion. This includes exploring methods to enhance scalability, optimize model performance, and potentially integrate additional sensors or modalities to achieve even greater detection accuracy. Our ultimate objective lies in continuously developing and innovating upon CNN-based techniques to minimize fall-related injuries and improve the quality of life for vulnerable geriatric individuals.

REFERENCES

[1] B. R. Greene, S. J. Redmond and B. Caulfield, "Fall Risk Assessment Through Automatic Combination of Clinical Fall Risk Factors and Body-Worn Sensor Data," in IEEE Journal of Biomedical and Health Informatics, vol. 21, no. 3, pp. 725-731, May 2017, doi: 10.1109/JBHI.2016.2539098.

[2] N. Bharathiraja, R. B. Indhuja, P. R. A. Krishnan, S. Anandhan and S. Hariprasad, "Real-Time Fall Detection using ESP32 and AMG8833 Thermal Sensor: A Non-Wearable Approach for Enhanced Safety," *2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*, Trichy, India, 2023, pp. 1732-1736, doi: 10.1109/ICAISS58487.2023.10250598

[3] J. G, P. S and S. D, "An Explainable Deep Learning Model for Vision-based Human Fall Detection System," *2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT)*, Kannur, India, 2022, pp. 1223-1229, doi: 10.1109/ICICICT54557.2022.9917979.

[4] R. R. Rajalaxmi, E. Gothai, S. V, S. Vignesh and T. Varun, "Vision based Fall Detection using Optimized Convolutional Neural Network," *2022 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, 2022, pp. 01-06, doi: 10.1109/ICCCI54379.2022.9740859.

[5] H. Sadreazami, M. Bolic and S. Rajan, "Fall Detection Using Standoff Radar-Based Sensing and Deep Convolutional Neural Network," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 1, pp. 197-201, Jan. 2020, doi: 10.1109/TCSII.2019.2904498.

[6] M. Saleh and R. L. B. Jeannès, "Elderly Fall Detection Using Wearable Sensors: A Low Cost Highly Accurate Algorithm," in *IEEE Sensors Journal*, vol. 19, no. 8, pp. 3156-3164, 15 April15, 2019, doi: 10.1109/JSEN.2019.2891128.

[7] Z. Zhang, X. Ma, H. Wu and Y. Li, "Fall Detection in Videos with Trajectory-Weighted Deep-Convolutional Rank-Pooling Descriptor," in *IEEE Access*, vol. 7, pp. 4135-4144, 2019, doi: 10.1109/ACCESS.2018.2887144.

[8] N. Zerrouki, F. Harrou, Y. Sun and A. Houacine, "Vision-Based Human Action Classification Using Adaptive Boosting Algorithm," in *IEEE Sensors Journal*, vol. 18, no. 12, pp. 5115-5121, 15 June15, 2018, doi: 10.1109/JSEN.2018.2830743.

[9] 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," in *IEEE Access*, vol. 6, pp. 9324-9335, 2018, doi: 10.1109/ACCESS.2018.2795239.

[10] S. Kim, M. Ko, K. Lee, M. Kim and K. Kim, "3D fall detection for single camera surveillance systems on the street," *2018 IEEE Sensors Applications Symposium (SAS)*, Seoul, Korea (South), 2018, pp. 1-6, doi: 10.1109/SAS.2018.8336746.

[11] K. Sehairi, F. Chouireb and J. Meunier, "Elderly fall detection system based on multiple shape features and motion analysis," *2018 International Conference on Intelligent Systems and Computer Vision (ISCV)*, Fez, Morocco, 2018, pp. 1-8, doi: 10.1109/ISACV.2018.8354084.

[12] M Ramkumar, K Jana, J Babu, VB Arunnachalam and SG Ashok, "Fall Detection System Using IoT" in In-data Intelligence and Cognitive Informatics, Singapore:Springer, pp. 315-323, 2022.

[13] B Priswanto and H Haryono, "Fall Detection using Sensors on a Smartphone", Sinkron: jurnal dan penelitian teknik informatika., vol. 7, no. 2, pp. 541-8, Apr 2022.

[14] S Nooruddin, MM Islam and FA Sharna, "An IoT based device-type invariant fall detection system", Internet of Things., vol. 9, pp. 100130, Mar 2020.

[15] MM Rahman, M Islam, S Ahmmed and SA Khan, "Obstacle and fall detection to guide the visually impaired people with real time monitoring", SN Compute Science., vol. 1, no. 4, pp. 1-0, Jul 2020.

[16] AB. Prajapati, "Person Fall Detection System Using Arduino UNO SIM900A and NEO6" in InRising Threats in Expert Applications and Solutions, Singapore:Springer, pp. 651-657, 2021.

[17] M Praneeth, MM Krishna, K Kavyalahari and VS. Kumar, "An IoT based intelligent fall detection and health monitoring system", Annals of the Romanian Society for Cell Biology, vol. 25, no. 6, pp. 17412-8, Jul 2021.

[18] SG Shoba, S Pradhan and MJ SonaliMetilda, "Development of Fall Detection and Alerting System for Elderly People Using Iot", Annals of the Romanian Society for Cell Biology., vol. 25, no. 6, pp. 2018-32, May 2021.

[19] A Tahir, W Taylor, A Taha, M Usman, SA Shah, MA Imran, et al., "IoT Based Fall Detection System for Elderly Healthcare" in Internet of Things for Human-Centered Design, Singapore:Springer, pp. 209-232, 2022.

[20] MH Sarowar, MF Khondakar, HS Roy, H Ullah, R Ahmed and QD. Hossain, "Internet of things-based fall detection and heart rate monitoring system for senior citizens", International Journal of Electrical and Computer Engineering (IJECE), vol. 12, no. 3, pp. 3204-16, Jun 2022.

[21] HC Chou and KY. Han, "Developing a smart walking cane with remote electrocardiogram and fall detection", Journal of Intelligent & Fuzzy Systems, vol. 40, no. 4, pp. 8073-86, Jan 2021