

Deep Learning-Based Near-Fall Detection Algorithm for Fall Risk Monitoring System Using a Single Inertial Measurement Unit

Ahnryul Choi, Tae Hyong Kim¹, Oleksandr Yuhai, Soohwan Jeong, Kyungran Kim, Hyunggun Kim², and Joung Hwan Mun³

Abstract—Proactively detecting falls and preventing injuries are among the primary keys to a healthy life for the elderly. Near-fall remote monitoring in daily life could provide key information to prevent future falls and obtain quantitative rehabilitation status for patients with weak balance ability. In this study, we developed a deep learning-based novel classification algorithm to precisely categorize three classes (falls, near-falls, and activities of daily living (ADLs)) using a single inertial measurement unit (IMU) device attached to the waist. A total of 34 young participants were included in this study. An IMU containing accelerometer and gyroscope sensors was fabricated to acquire acceleration and angular velocity signals. A comprehensive experiment including thirty-six types of activities (10 types of falls, 10 types of near-falls, and 16 types of ADLs) was designed based on previous studies. A modified directed acyclic graph-convolution neural network (DAG-CNN) architecture with hyperparameter optimization was proposed to predict fall, near-fall, and ADLs. Prediction results of the modified DAG-CNN structure were found to be approximately 7% more accurate than the traditional CNN structure. For the case of near-falls, the modified DAG-CNN demonstrated excellent prediction performance with accuracy of over 98% by combining gyroscope and accelerometer features. Additionally, by combining acceleration and angular velocity the trained model showed better performance than each model of acceleration and angular velocity. It is believed that information to preemptively handle the risk of falls and

quantitatively evaluate the rehabilitation status of the elderly with weak balance will be provided by monitoring near-falls.

Index Terms—Pre-impact fall detection, near-fall detection, convolution neural network, directed acyclic graph, inertial measurement unit.

I. INTRODUCTION

BALANCED posture is the process of securing an equilibrium state by controlling the deviation of the center of gravity within the base of support of the human body [1]. It is maintained through the integration of the visual, vestibular and somatosensory systems [2], [3], [4]. Although the definition and causes of falls vary [5], from the biomechanical perspective, falls can be described as a phenomenon where the center of gravity drops rapidly toward the ground and parts of the human body hit the ground. Every year, 32% of elderly people over the age of 70 experience fall incidents, and more than half of those who have been injured show a high case fatality rate within 6 months of the fall [6], [7]. Elderly people who have had fall incidents recover slowly even with slight bruising or simple fractures, leading to declining physical independence and pathologic conditions in activity [8]. Therefore, a primary key to a healthy life in the elderly is to proactively detect falls and prevent injuries.

Fall monitoring systems utilize wearable sensors such as accelerometers and gyroscopes [9]. These sensors have gradually become smaller and lighter and have been designed to attach to the human body easily. In particular, a single inertial measurement unit (IMU), a hardware system made of an accelerometer and an angular velocity sensor, enables low-cost, low-complexity, low-power, and high-speed data processing, and is widely used for various analyses including motion analysis and fall detection [10]. Most of the fall detection tests are focused on falls and activities of daily living (ADLs). Falls can be divided into four phases in terms of fall detection: pre-impact, impact, unstable, and stable phases. Many researchers have used the characteristics of the impact and stable phases to describe falls [11]. This type of fall detection method is called post-impact fall detection, which can determine whether elderly people have fallen but cannot prevent injuries from falls. Alternatively, pre-impact fall detection attempts to find the moment before fall. Once a pre-impact fall is detected, an airbag system or inflatable hip protection can be activated

Manuscript received 3 May 2022; revised 24 July 2022; accepted 11 August 2022. Date of publication 16 August 2022; date of current version 1 September 2022. This work was carried out with the support of “Cooperative Research Program for Agriculture Science and Technology Development (Project No. PJ01531103)” Rural Development Administration, Republic of Korea. (Corresponding authors: Hyunggun Kim; Joung Hwan Mun.)

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Ethics Committee of Sungkyunkwan University.

Ahnryul Choi is with the Department of Biomedical Engineering, Catholic Kwandong University, Gangneung, Gangwon 25601, South Korea, and also with the Department of Bio-Mechatronic Engineering, Sungkyunkwan University, Suwon, Gyeonggi 16419, South Korea (e-mail: achoi@cku.ac.kr).

Tae Hyong Kim, Oleksandr Yuhai, Soohwan Jeong, Hyunggun Kim, and Joung Hwan Mun are with the Department of Bio-Mechatronic Engineering, Sungkyunkwan University, Suwon, Gyeonggi 16419, South Korea (e-mail: sanctified@skku.edu; oleksandr@g.skku.edu; jeongsoohwan92@gmail.com; hkim.bme@skku.edu; jmun@skku.edu).

Kyungran Kim is with the Agricultural Health and Safety Division, Rural Development Administration, Jeonju, Jeollabuk 54875, South Korea (e-mail: kimgr@korea.kr).

This article has supplementary downloadable material available at <https://doi.org/10.1109/TNSRE.2022.3199068>, provided by the authors.

Digital Object Identifier 10.1109/TNSRE.2022.3199068

to significantly reduce injuries for the elderly [12], [13]. However, pre-impact fall detection remains a tough challenge as the free-fall phase is not clear. Regarding experimental equipment for pre-impact fall detection, only a few studies have used wearable IMU devices [12], [14], [15].

One recent strategy for pre-impact fall detection is to conduct near-fall detection [9]. Near-fall is a phenomenon in which balance is lost but a correct recovery operation actually leads to not falling [16]. Elderly people experience multiple near-falls before experiencing falls and steadily monitoring near-falls in their daily lives can help predict falls in advance [17]. Near-falls are included in the non-fall category as there is no actual fall, but they have features that distinguish them from ADLs (hit, bump, slip, trip, and misstep) [18]. For example, near-falls have various other types of data including stumble, incorrect weight transfer, and leaning too far in any direction [19]. It is imperative to conduct near-fall detection for more accurate and more practical fall risk assessment [20]; therefore, near-fall remote monitoring in daily life could provide key information to prevent future falls, as well as quantitative rehabilitation status for patients with weak balance ability [17]. Moreover, near-fall detection can reduce various malfunctions of protection equipment including airbag systems and inflatable hip protection gears. However, experimental analysis of near-fall detection systems has not been conducted as much as fall detection since near-fall is not as specific a situation as fall when monitoring daily life of the elderly. The classification of fall, near-fall, and ADLs is a challenge.

Previous fall detection studies often employed threshold-based methods because they are easily to apply. Accelerometers and gyroscopes are commonly used to understand the characteristics and patterns of acceleration and angular velocity and to determine the thresholds [14], [21]. More recently, machine learning-based methods have been utilized to classify falls. Feature points can be selected from the accelerometers and gyroscopes using classical machine learning methods such as support vector machine (SVM), k-nearest neighbor (KNN), or deep learning algorithms such as artificial neural network (ANN) and long short-term memory (LSTM) [12], [21], [22], [23]. Classification of fall detection using machine learning algorithms shows higher performance than the threshold-based classification, and the recent advances in deep learning-based algorithms particularly enables even more precise classification in various fields [11], [24]. Conventional fall detection methods have mainly performed the classification of fall and ADLs due to the difficulty of near-fall detection (i.e., pre-impact fall detection). While there are a few cases available to employ various near-fall data (suppl. Table I) [8], [9], [25], [26], [27], [28], [29], near-fall detection is an important factor for a comprehensive understanding of fall mechanism since repetitive long-term near-falls would lead to fall incidence [25].

Various studies to estimate a fall using an IMU device so far have attempted to discriminate between falls and ADLs. Even studies on detecting near-falls have been limited to a binary classification (fall vs. near-fall or near-fall vs. ADLs) using single or a few kinds of near-fall data [17], and there is no study of comprehensively classifying falls, near-falls, and ADLs utilizing various types of fall, near-fall, and

ADLs experimental data. Because near-falls occur in various situations that can overlap with actual falls or ADLs, it is challengeable to classify near-falls along with falls and ADLs. However, precisely classifying near-falls along with falls and ADLs can provide a clue to the risk of falls in the elderly or vertigo patients in advance. Additionally, the quality of the fall prediction system can be improved by decreasing false alarms through accurate detection of near-falls. In the present study, we developed a novel deep learning-based classification algorithm to precisely categorize three classes (fall, near-fall, and ADLs) using a single IMU device attached to the waist for the application to a hip protection system. Our specific objectives were as follows:

1. Custom design and manufacture a single IMU hardware device that can acquire acceleration and angular velocity signals.

2. Establish a novel pre-impact fall detection algorithm based on a deep learning model that can precisely classify falls, near-falls, and ADLs using comprehensive experimental data (10 types of falls, 10 types of near-falls, and 16 types of ADLs).

3. Evaluate the performance between deep learning models (modified directed acyclic graph CNN vs. typical CNN) and between using sensor signals (acceleration vs. angular velocity vs. combination of acceleration and angular velocity) of the IMU device.

The rest of this paper is organized as follows: Section II presents the materials and methods of data collection, proposed prediction procedure and performance measure. The experimental results with the proposed deep learning model are demonstrated in section III. Finally, we interpret the results, state the contribution of this study and give the limitations and future direction of the study. Additionally, we have provided a literature review related to near-fall detection as supplementary material.

II. MATERIALS AND METHODS

A. Data Collection

- 1) *Subjects*: A total of 34 young participants (21 males and 13 females) aged between 21 and 34 years (mean = 27.6 years), with a body mass of 45 - 81 kg (mean = 62.1 kg), and a height from 1.57 to 1.85 m (mean = 1.67 m), participated in this study. All participants were healthy and did not have any type of agnosia disease or musculoskeletal problems. Prior to the experiments, all participants filled out consent forms. The entire experiments were carried out in accordance with the relevant approved guidelines of the Ethics Committee of Sungkyunkwan University.

- 2) *Apparatus*: An inertial measurement unit (IMU) containing 3-axis accelerometer and 3-axis gyroscope sensors was fabricated to acquire acceleration and angular velocity signals of falls, near-falls and ADLs. An IMU module (MPU-6050, JK Electronics, Seoul, South Korea) with an electricity consumption with a voltage of 3 - 5 V was used in this device. The fabricated device contained a 16-bit analog-to-digital converter for simultaneous sampling of the 3-axis accelerometer and gyroscope. The data output range was $\pm 250 - 2,000$ d/s for the gyroscope and $\pm 2 - 16$ G for the accelerometer. The data were collected at a sample rate of 40 Hz. A Bluetooth V4.0 BLE

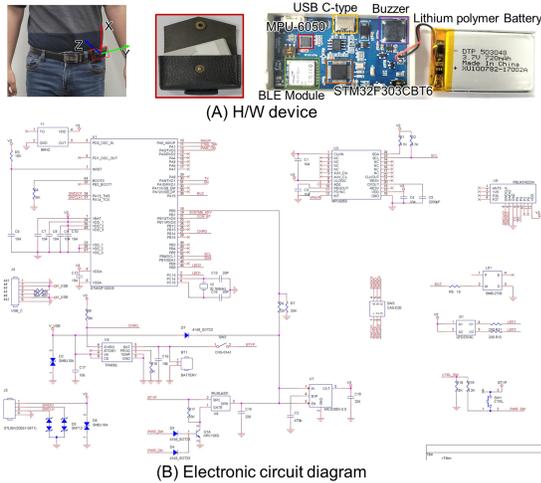


Fig. 1. (A) IMU hardware device and (B) electrical circuit diagram.

(HM-11, JN Huamao Technology Co., Jinan, China) module was used for wireless connection between the sensing and data collection devices. A microcontroller (STM32F103CBT6, STMicroelectronics, Geneva, Switzerland) with a 3.7 V / 550 mAh lithium polymer battery (DTP 503040-PCM, Shenzhen Data Power Technology Ltd., Shenzhen, China) was used for data collection. The size of the device (including the protective plastic case) was 60 (length) \times 35 (width) \times 15 (height) mm, and the weight was 150 g. The device was connected to a workstation PC using the Bluetooth module and a Bluetooth signal receiver. C#-based Windows form data acquisition software (HMSoft, JN Huamao Technology Co., Jinan, China) was utilized to collect real-time 3-axis accelerometer and gyroscope data. A detailed hardware and electrical circuit diagram of the device is shown in Fig.1.

3) Experimental Protocols: First, each participant was asked to watch a pre-recorded video clip containing a detailed demonstration of each activity. For each trial, the participants were asked once again to watch the segment of the video clip for the corresponding action. The custom-designed IMU sensor was positioned on the left anterior iliac crest of the pelvis [10]. The workstation PC and Bluetooth signal receiver were located at the side of the protective mat.

Each activity experiment attempt was made to implement active daily livings, fall, and near-fall scenarios as realistically as possible. Ten classes of near-falls were chosen based on previous near-fall-related studies: near-fall forward due to tripping, near-fall backward due to tripping, near-fall forward due to slipping, near-fall backward due to slipping, near-fall forward due to misstep, near-side fall on the right side, near-side fall on the left side, sit-to-stand near-fall forward, hit/bump near-fall forward, and hit/bump near-fall backward [9], [27]. For the safety of subjects, a gymnasium mattress and high-density ethylene foam were placed where subjects might fall to reduce the impact force during fall or near-fall movements. Subjects were also asked to wear knee and elbow protective gears and a helmet [10]. For each near-fall action, the subjects had to by any means restore balance on their own (e.g., to shift their center of mass by certain movements of their arms and legs) as demonstrated in the

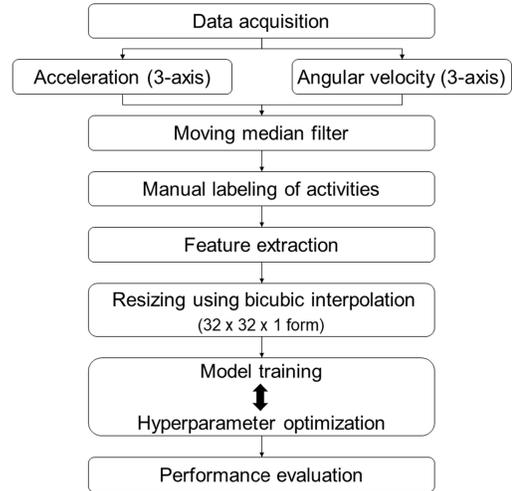


Fig. 2. Flowchart of the proposed procedure to discriminate between falls, near-falls, and ADLs.

pre-recorded demo video. Before simulating active events such as near-fall due to tripping, slipping, misstep, and walking to side near-fall, the subjects were asked to walk 3 m from the specified initial position at a self-preferred speed. In the case of tripping near-fall, we tightly fixed a plastic box of size 32 (length) \times 23.5 (width) \times 9 (height) cm to the floor with packing tape at 15 cm from the safety mat. For slipping near-fall, we utilized a thin polyvinyl chloride (PVC) desk pad of size 81 (length) \times 50 (width) \times 0.2 (height) cm, placed closely to the safety mat and firmly fixed at the edges with packing tape to the floor. Before each slipping near-fall trial, we wet this PVC pad using a water sprayer to minimize friction and create a more realistic slipping condition [20]. For passive events such as sit-to-stand near-fall and hit/bump near-fall, the subject had to take the correct initial position (sit-to-stand fall: sit comfortably on a chair located 40 cm from the mat, hit/bump near-fall: stand at the initial position located 10 cm from the mat), and stay still in this pose for 2-3 sec. After completing each attempt, the subject had to stop at the marked spot 15 cm from the mat (if the action ended inside the mat, the subject had to walk slowly to the indicated place) and stand motionless for 3-5 sec before returning to the initial position. The 16 classes of ADLs (standing, forward lying, backward lying, leftward lying, rightward lying, sitting, sit to stand, stand to sit, stand to lying, lying to stand, walking, walking by bending, jumping, walking to jumping, upstairs and downstairs) and the 10 classes of falls (forward fall, backward fall, leftward fall, rightward fall, fall while standing, sitting on empty chair, forward fall while walking, backward fall while walking, tripping forward and slipping backward) were designed based on the previous studies [10], [30], [31]. We collected raw data containing 680 occasions of near-falls, 2,176 of ADLs, and 1,360 of falls.

B. Proposed Prediction Procedure for Falls, Near-Falls, and ADLs

Our proposed pre-impact fall detection model with ADLs and near-falls is described in Fig. 2. The flow consists of data acquisition (3-axis acceleration and angular velocity),

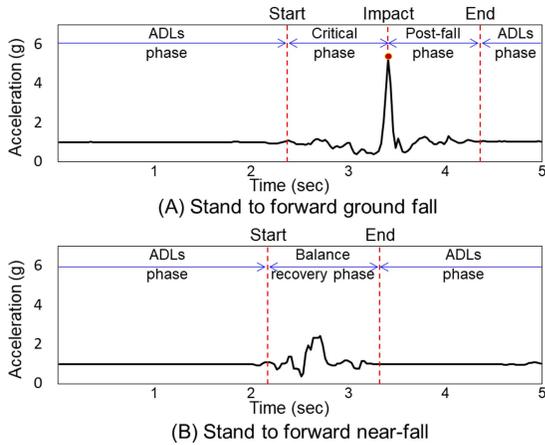


Fig. 3. Comparison between fall (stand to forward ground fall) and near-fall (stand to forward near-fall).

preprocessing (filtering and labeling), feature extraction, model training with hyperparameter optimization and performance evaluation. More detailed descriptions are as follows.

1) *Fall and near-fall*: Basically, there are four phases of fall. The pre-fall phase is when normal activity occurs (classified as ADLs) prior to the body losing balance. Next, the critical phase is the time when the fall begins. The impact occurs when the subject contacts the ground due to the fall. Lastly, during the post-fall phase, the subject lies entirely on the surface of the fall. The complete end of the post-fall phase and the return of the subject to the standing position is marked as “End” (Fig. 3(A)). We pay attention to the detection of fall impact acceleration and angular velocity magnitude prior to the impact point within the critical phase [13], [14]. In this way, it becomes possible to detect falls (and their various types) before the moment of impact while labeling the data activity type.

In the case of near-fall, there are three main phases similar to fall: the pre-near-fall phase (classified as ADLs) occurs before the body loses balance. The start of the balance recovery phase is the moment when near-fall begins due to various factors such as tripping and slipping. During the balance recovery phase, the subject performs various movements to shift the arms and legs and restore the center of mass to the normal and stable state to prevent an actual fall [20]. The completion of the balance recovery phase and the return of the subject to the stable standing position are marked as “End” (Fig. 3(B)). In this study, the critical and balance recovery phase to detect among falls, near-falls and ADLs are labeled as the fall and near-fall states, respectively.

C. Data Preprocessing

The data obtained from the IMU sensors were thoroughly reviewed for incorrect and duplicate data, which could cause inadequate operation of the machine learning algorithm. To improve the quality of the dataset, we identified and removed the various errors and inconsistencies in the collected data resulting from the loss of the Bluetooth signal between the device and workstation PC. A sliding median filter was used to remove noise and obtain more reliable datasets. One of the key operations in data preprocessing to develop a

machine learning-based classification or prediction model is data labeling [32]. In this study, we manually labeled each data frame using the calculated sum magnitude vector (SMV) of the accelerometer signal to accurately identify and label the signal sections responsible for the ADLs, fall (critical phase), and near-fall (balance recovery phase), respectively. Lastly, we set a window size of 40 and divide the signal by using an overlapping window size of 50% [10].

D. Feature Extraction

Feature extraction is one of the most important steps for human activity recognition (HAR) models. Building an efficient feature vector improves the performance of machine learning algorithms. Kim *et al.* [10] selected 40 specific features to achieve the best performance of their machine learning-based HAR algorithms. Considering previous HAR research using accelerometers and gyroscopes, we selected 40 unique features to effectively detect ADLs, falls, and near-falls [33], [34]. In this study, we extracted 23 main accelerometer features and 7 gyroscope features, excluding the raw signal from both sensors. In addition, we obtained the estimation of the systemic quaternion to approximate the orientation of the subject using the extended Kalman filter (Quaternions) [9]. The 40 features were calculated as shown in Table I. Therefore, a squared input matrix for the deep learning model was created with size of 40 (window size) \times 40 (number of features).

1) *Deep Learning Network*: Typical CNN models have a hierarchical chain structure with feedforward connection through which various invariant representations of the input data are calculated, i.e., certain features are extracted [35].

However, CNN models do not consider the middle- and low-level features, which can play important roles in the classification of ADLs, falls, and near-falls, especially in the case of a multiclassification problem. Therefore, we proposed a novel deep neural network architecture, namely modified directed acyclic graph convolutional neural network (DAG-CNN) (Fig. 4). This architecture has the advantage of relatively efficient, simple, and fast detection and classification of near-falls by extracting various multilevel features from the deep layers of the CNN. DAG-CNN allows us to extract not only high-level features, but various multi-level features by adding additional “branched layers”. The process of training the main CNN structure and its branches occurs simultaneously, reducing the size of the model and quickly and more efficiently completing the process of feature extraction [36].

The squared input matrix is created by calculating features. A bicubic interpolation was then employed to resize the input matrix obtained from data pre-processing into a suitable input data size for our proposed DAG-CNN that is 32 (height) \times 32 (width) \times 1 (channel). The main structure of our proposed DAG-CNN model starting with the input layer with zero center normalization using the formula below:

$$Z = \frac{x - \mu}{\sigma} \quad (1)$$

where x is the original data, μ is the mean, and σ is the standard deviation.

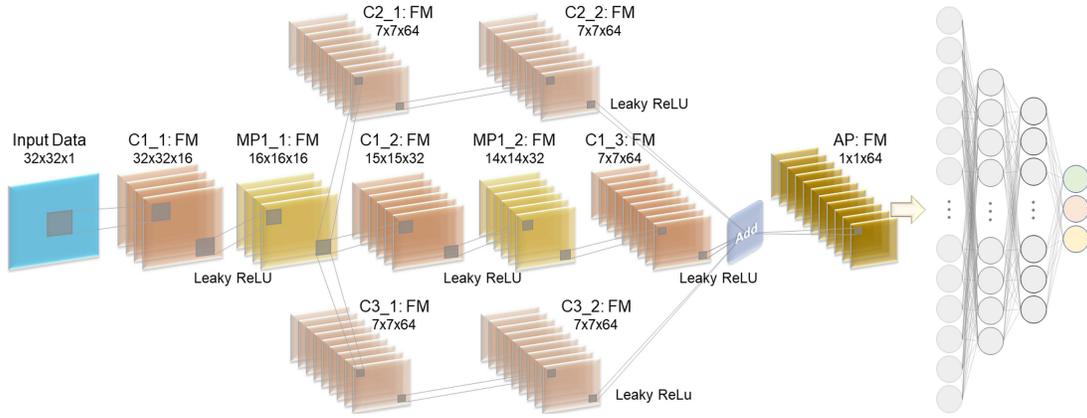


Fig. 4. Architecture for modified DAG-CNN model.

TABLE I
TYPES OF FEATURES EXTRACTED FROM THE IMU DEVICE

| | Accelerometer | Gyroscope | Combination |
|----|--------------------------------|-----------------------------|----------------------------|
| 1 | Raw acceleration (x) | Raw angular velocity (x) | All accelerometer features |
| 2 | Raw acceleration (y) | Raw angular velocity (y) | All gyroscope features |
| 3 | Raw acceleration (z) | Raw angular velocity (z) | Quaternion (x) |
| 4 | Mean (x, y, z) | Mean (x, y, z) | Quaternion (y) |
| 5 | Variance (x, y, z) | Sum magnitude vector | Quaternion (z) |
| 6 | Standard deviation (x, y, z) | Min. (x, y, z) | Quaternion (w) |
| 7 | Sum magnitude area | Max. (x, y, z) | |
| 8 | Sum magnitude vector | Difference of max. and min. | |
| 9 | Sum vector in horizontal plane | Interquartile range | |
| 10 | Tilt angle (x, y) | Median (x, y, z) | |
| 11 | Sagittal angle (y, z) | | |
| 12 | Min. (x, y, z) | | |
| 13 | Max. (x, y, z) | | |
| 14 | Difference of max. and min. | | |
| 15 | Interquartile range | | |
| 16 | FFT (x) | | |
| 17 | FFT (y) | | |
| 18 | FFT (z) | | |
| 19 | Sum of FFT | | |
| 20 | Power of FFT (x) | | |
| 21 | Power of FFT (y) | | |
| 22 | Power of FFT (z) | | |
| 23 | Sum of power of FFT (x, y, z) | | |
| 24 | Jerk (x) | | |
| 25 | Jerk (y) | | |
| 26 | Jerk (z) | | |

Next, it consists of three convolutional layers with kernel sizes of 5×5 , 2×2 , and 2×2 and with 16, 32, and 64 filters, respectively. We employed Leaky ReLU with a standard parameter (0.01) as an activation function for each convolutional layer of the main structure. The Leaky ReLU can be calculated as follows:

$$f(x) = \begin{cases} 0.01x, & \text{if } x < 0 \\ x, & \text{otherwise} \end{cases} \quad (2)$$

where x is the element of the output feature map from the previous convolutional layer [37].

Following the first two activation functions (convolutional and leaky ReLU layers), MaxPooling was applied to reduce computational complexity and variance and extract the most important low-level features from the neighborhood of the elements of each feature map. The size of the feature map after pooling (maximum or average) can be calculated using the following formula in equation 3.

$$\left\lceil \frac{I - P}{S} \right\rceil + 1 \quad (3)$$

where I is the input data shape, P is the pooling window size, and S is the size of stride.

Branched structures (the top and bottom layers in Fig. 4) consist of two consecutive convolutional layers, each of which has filters of size 3×3 . The first stacked convolutional layer has a stride size of 2×2 with no pooling, while the second stacked convolutional layer has no stride with a pooling size of 1×1 . The stacked convolutional layers (without pooling layer between the convolutional layers) of these branched structures allow features to be transferred and learned directly from the first convolutional layer to the second without losing any data. It also allows a hierarchical decomposition of the input and provides more flexibility in expressing non-linear transformations without losing information. Hence, it is important to note that features of a different level and type, contrasting from the main structure features, are extracted in these branched structures. These branched structures are then attached to the “Add” layer, where the element-wise addition of two branched function maps to the characteristic map of the main structures is performed as in the formula below.

$$\begin{bmatrix} a & \dots & b \\ \vdots & \ddots & \vdots \\ c & \dots & d \end{bmatrix} + \dots + \begin{bmatrix} e & \dots & f \\ \vdots & \ddots & \vdots \\ g & \dots & h \end{bmatrix} = \begin{bmatrix} a + \dots + e & \dots & b + \dots + f \\ \vdots & \ddots & \vdots \\ c + \dots + g & \dots & d + \dots + h \end{bmatrix} \quad (4)$$

Prior to the classification layer, average pooling on the feature map obtained from the last add layer is applied. In this way, we performed the generalization of feature maps from all channels (the main structure and two branched structures)

TABLE II
OPTIMIZED HYPERPARAMETERS OF THE CNN AND MODIFIED
DAG-CNN MODELS

| | Training option | Range of parameters | CNN | Modified DAG-CNN |
|---|---------------------------|--------------------------|---------|------------------|
| 1 | Initial learning rate | [1e-6, 1e-2] | 0.0004 | 0.001 |
| 2 | L2 regularization | [1e-10, 1e-2] | 3.4e-08 | 0.002 |
| 3 | Optimizer | [Sgdm, Adam] | Adam | Adam |
| 4 | Mini batch size | [8, 16, 32] | 32 | 32 |
| 5 | Maximum epochs | [10, 20, 30, 40, 50, 60] | 40 | 40 |
| 6 | Learning rate drop period | - | 10 | 10 |
| 7 | Learning rate drop factor | - | 0.2 | 0.2 |

of our modified DAG-CNN without losing any important information or features.

For a nonlinear combination of the data of multi-level features and its classification, the output data of the convolutional layers representing the features of several levels pass through the final average pooling layer to a fully connected deep CNN model consisting of three fully connected layers. The first layer consists of 256 neurons, the second layer contains 128 neurons, and the third layer has 64 neurons. To optimize the model, enhance its speed, and prevent possible overfitting problems, a dropout with a parameter of 20% is used after the first layer with 256 neurons to randomly remove 20% of the neurons. Next, the SoftMax function was used to predict the polynomial probability distribution between the multiple classes. The last fully connected layer number of our proposed DAG-CNN-based deep learning model was set to be 3 nodes to classify into three classes of activities (1: falls, 2: near-falls, and 3: ADLs). Detailed architecture and parameters of DAG-CNN are listed in suppl. Table II. In order to evaluate the performance of the proposed DAG-CNN architecture, we compared it with a classic LeNet-5 CNN model [38]. The typical CNN architecture is made of input layer, 3 convolution layers with leaky ReLU layer and max pooling layer between each convolution layer and average global pooling layer. Three convolution layers utilized kernel size of 5×5 , 2×2 and 2×2 with the number of output filter size of 16, 32 and 64, respectively. Next, three fully connected layers and a softmax layer is followed. The general architecture of the typical CNN model is similar to the main branch of the proposed DAG-CNN architecture (excluding layers 12 to 18 in suppl. Table II). MATLAB (2020b, Mathwork, USA) was utilized for training and testing CNN models with a window system of Intel(R) i7-5930K CPU @ 3.50GHz, Nvidia Titan XP GPU 12GB.

2) *Hyperparameter Optimization*: Optimal hyperparameter values for training deep learning and CNN architecture setting for best performance achievement while preventing from overfitting was found using Bayesian optimization technique. Bayesian optimization reflects prior knowledge to find optimal values during a search process. This technique finds the best value based on two elements: (i) surrogate model and ii) acquisition functions [39]. The objective function for this study is to minimize the classification error of near-fall detection model. Five parameters (Initial learning rate, L2 regularization

value, optimizer type, mini batch size, maximum epochs value) for training were selected as design parameters and their lower and upper bound values are listed as shown in Table II. The optimal parameters for the learning rate drop period and learning rate drop factor were set manually. Maximum objective evaluation number was set as 20 and the best hyperparameter values were used to train our deep learning model as final training.

E. Performance Measure

In this study, we applied leave-one-participant-out cross-validation techniques to validate our model [40]. The performance criteria of accuracy, specificity, sensitivity, precision, Negative Predictive Value (NPV), F1-score, and Matthews Correlation Coefficient (MCC) were calculated for each of the three classes (ADLs, falls and near-falls) separately, and then their averages were used to further compare the two deep learning models (suppl. Table III). Specifically, MCC is a more reliable statistic compared to accuracy, that gives a high score only if the prediction performs well in all four categories of the confusion matrix (true positive, false negative, true negative, and false positive), proportionally to the size of the positive elements and the size of the negative elements in the dataset at the same time. It returns a value between -1 and +1, where +1 represents a perfect prediction, 0 is no better than a random prediction, and -1 indicates a complete mismatch between prediction and observation. It can be calculated using the following formula.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (5)$$

where, TP, TN, FP, and FN mean true positive, true negative, false positive, and false negative, respectively.

III. RESULTS

The training and validation process of accuracy curves are displayed in Fig. 5. The modified DAG-CNN converged closely to 100% accuracy, but the accuracy of the CNN converged to approximately 93%. The accuracy of the training process for both models became gradually stable after the 20th epoch and hardly altered after the 30th epoch. Therefore, to prevent overfitting, we stopped the training process at the 40th epoch. The modified DAG-CNN also demonstrated a faster convergence speed than CNN.

Confusion matrices of CNN and modified DAG-CNN model by using gyroscope, accelerometer and combination of gyroscope and accelerometer features are shown in Fig. 6. The gyroscope feature-based CNN showed low performance with an accuracy range between 69.8% and 89.7%. However, the modified DAG-CNN demonstrated excellent prediction performance with an accuracy of over 98% for combination of gyroscope and accelerometer features. For the detection of ADLs and falls, the use of the accelerometer features showed better performance than the use of the gyroscope features in both CNN and modified DAG-CNN models. However, for near-fall detection alone, the use of the gyroscope features

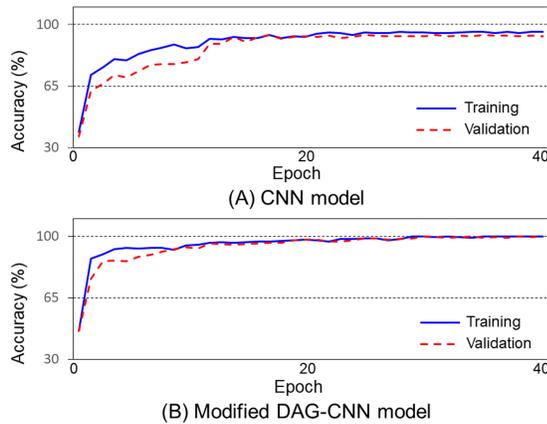


Fig. 5. Training and validation accuracy of CNN and modified DAG-CNN models.

showed superior performance to the use of the accelerometer features in both CNN and modified DAG-CNN models as shown in Fig. 6(A), 6(B), 6(D) and 6(E). The performance difference between the uses of the accelerometer and gyroscope features with the CNN confusion matrix was greater than that with the modified DAG-CNN confusion matrix. Both CNN and modified DAG-CNN models based on the combination of accelerometer and gyroscope features showed better performance compared to those based on accelerometer and gyroscope alone.

Based on the confusion matrices of CNN and modified DAG-CNN models, the receiver-operating characteristic (ROC) curves are displayed and computed area under curve (AUC) for three classes (ADLs, fall and near-fall) as shown in Fig. 7. The best performance was obtained from the modified DAG-CNN using the combination of the accelerator and gyroscope features, with a sharp decreasing true positive rate with decreasing false positive rate. The ROC curve of CNN only with the gyroscope features showed a better near-fall detection performance than ADLs and falls (Fig. 7(A)). Similarly, the modified DAG-CNN only with the gyroscope features showed high performance in near-fall detection (Fig. 7(D)). For the CNN model, the AUCs of the near-fall class for the gyroscope, accelerometer and combination of accelerometer and gyroscope feature-based models were 0.97, 0.93 and 0.98, respectively. The AUCs of ADLs and falls were higher than near-falls for the accelerometer feature-based CNN model. However, the AUC of near-falls was higher than ADLs and falls for gyroscope and combination of accelerometer and gyroscope feature-based model as shown in Fig. 7(A) and 7(C). As shown in Fig. 7(D) to 7(F), the AUCs of near-fall for three different feature-based models were 0.99, 0.99 and 0.99, respectively. The AUCs of near-fall for gyroscope showed higher values than AUCs of ADLs and falls.

The algorithm working time (system latency) and lead time (time difference between the time of fall impact and pre-fall detection) are presented in Table III. The system latencies of fall, near-fall, and ADLs were about 56.4 and 58.5 ms for the CNN and modified DAG-CNN models, respectively, with similar computation times. The average lead time was presented as 660.9 and 662.0 for the CNN and modified DAG-CNN models, respectively. The inflation time of the protection systems, such as airbags, will require about 120 ms [41]. Even considering

the system latency, enough inflation time for the protection system could be acquired in future work.

The overall quantitative performance evaluation of CNN and the modified DAG-CNN is demonstrated in Fig. 8. The modified DAG-CNN outperformed CNN in every comparison of performance criterion. The accuracies of CNN and the modified DAG-CNN only with the gyroscope features ranged between 70.4% to 86.8% and 80.9% to 92.2%, respectively. The average accuracy and F1-score with the accelerometer features showed an improvement in the modified DAG-CNN with a value of 96.6% and 0.96 compared to CNN with a value of 90.0% and 0.88, respectively. The combination of gyroscope and accelerometer feature-based model for CNN and modified DAG-CNN showed improvement of 14% and 10%, respectively, for F1-score as compared to gyroscope feature-based model.

IV. DISCUSSION

Near-falls, which exist between falls and ADLs, generally occur more frequently than practical falls [17]. Any system that can automatically quantify the frequency of near-falls can complement the existing self-report to monitor balance status and evaluate fall risk more precisely [28]. In this study, we proposed a deep learning-based precise fall detection algorithm capable of simultaneously detecting fall, near-fall, and ADLs. In the modified DAG-CNN model, fall, near-fall and ADLs were classified with accuracy of up to 99%. In addition, when the model was trained using features based on the combination of acceleration and angular velocity data, the accuracy was improved by about 3% and 12% compared to when acceleration and angular velocity were used, respectively.

One of the key problems of pre-impact fall detection algorithms is the lack of available indicators directly related to dynamic near-fall detection [25]. It is also very difficult to treat near-falls and falls because some active types of near-falls produce higher acceleration peaks than falls, which was demonstrated using the public dataset of Cogent Lab [42]. In contrast, the median value of the minimum SMV of accelerometer signal for near-falls is often lower than that for falls during the free-fall phase. In addition, Lee *et al.* [9] experimentally demonstrated that various balance recovery movements during simulated near-falls can lead to significant increase in acceleration peak, which becomes a challengeable constrain in recognizing falls and near-falls. On the other hand, Trkov *et al.* [43] demonstrated that the extended Kalman filter-based prediction for cases of small slips can have a similar nature to walking through dynamic analysis (neglecting the friction force) of slips of different intensities. It is also important to note that the lack of a certain homogeneity between the various types of near-falls such as trip, slip, misstep, etc., makes identifying near-falls based on IMU data alone, without contextual information, an especially difficult task [28]. In our study, almost all possible types of falls, near-falls, and ADLs based on the previous fall-related studies were included from the comprehensive experiments, taking into account the problem of data heterogeneity depending on various factors such as severe types of near-falls almost similar to the fall patterns and relative passive types of near-falls similar to the ADLs in nature. Despite the heterogeneity

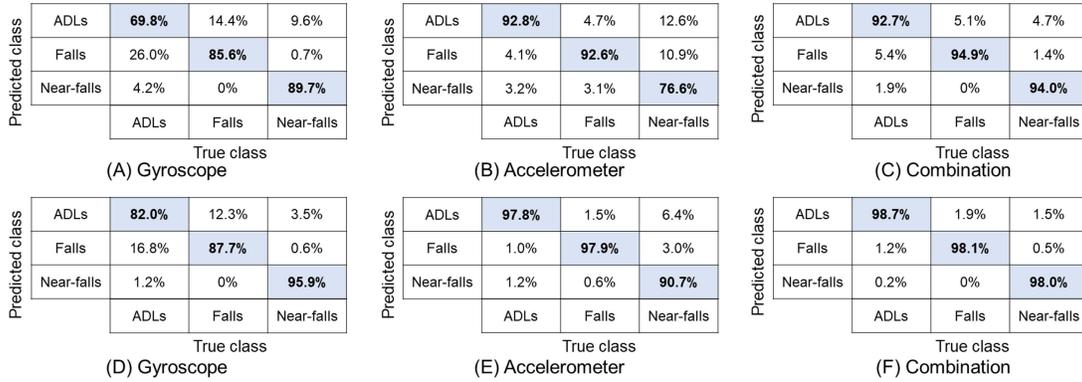


Fig. 6. CNN confusion matrix ((A) gyroscope, (B) accelerometer, and (C) combination of gyroscope and accelerometer) and modified DAG-CNN confusion matrix ((D) gyroscope, (E) accelerometer, and (F) combination of gyroscope and accelerometer).

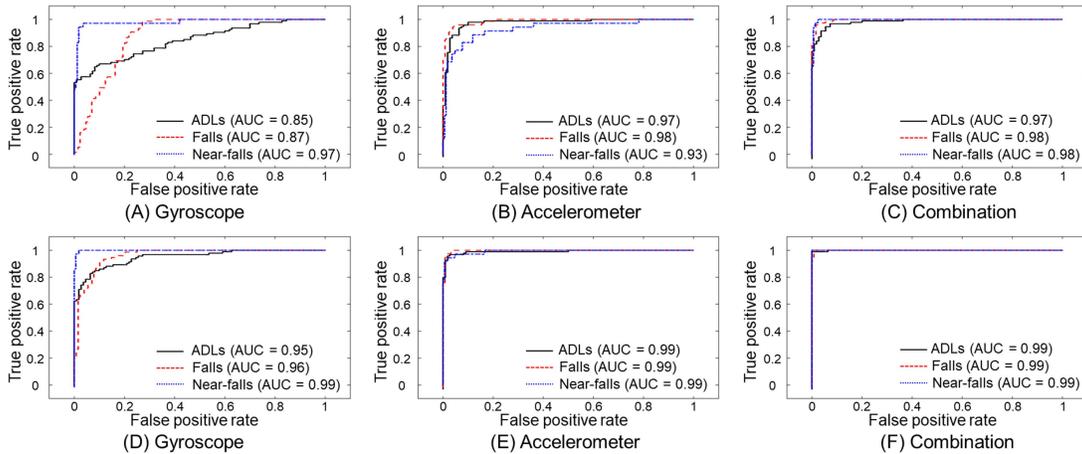


Fig. 7. CNN ROC curves ((A) gyroscope, (B) accelerometer, and (C) combination of gyroscope and accelerometer) and modified DAG-CNN ROC curves ((D) gyroscope, (E) accelerometer, and (F) combination of gyroscope and accelerometer).

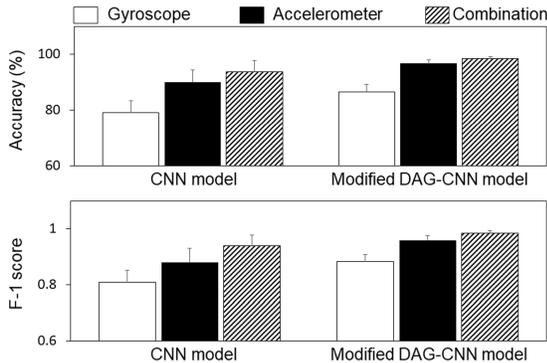


Fig. 8. Accuracy and F-1 score of leave-one-participant-out cross-validation from CNN and modified DAG-CNN models.

characteristics of near-fall data and various abnormal artifacts of movement arising from the relatively strong relocation of the center of gravity to restore balance, there is a strong need to categorize falls, near-falls, and ADLs as three distinct classes.

In this study, modified DAG-CNN architecture that can increase the dimension of input features was used to discriminate between falls, near-falls, and ADLs (Fig. 4). A classical straightforward CNN (3 convolutional layers, 2 subsampling layers and 2 fully connected layers) is similar to a structure of feed-forward neural network. Each convolutional layer is made of convolution, pooling and nonlinear activation functions. Usually, an average pooling layer and tanh activation function

are applied with multi-layer perceptron as last classifier. This type of model extract features fits to coarse classification not fine-graded ones. For instance, the walking and standing activity shows discrete signal difference which can be a coarse classification. DAG-based network shows characteristics of no circular cycle with one-direction presented as topologically sorted. It was explored earlier in the context of recurrent neural networks that use feedback from forward layers to backward layers to capture dynamic states. This type of network is widely applied in blockchain field such as Byteball coin [44]. DAG-network has high transaction speed since each unit is processed by parallel and relatively small network size. This type of structure allows to directly learn multi-scale representations (features) and fuse multi-scale features into a single training procedure [45]. Previous research reported that DAG-CNN network can discriminate with higher detection performance since it can extract multi-scale features from multiple layers simultaneously from coarse to fine features [46]. In this study, the feature-level fusion [47], which could combine extracted features from different levels (high, middle, low) and create the final feature vector, was employed. It provided more discriminative features for classifying fall, near-fall and ADLs. A previous study reported that the baseline chain structured CNN and DAG-network base CNN showed increased detection rate range between 8% to approximately 20%, which is similar to our 3-class prediction rate [45].

TABLE III
SYSTEM LATENCY AND LEAD TIME OF THE CNN AND MODIFIED DAG-CNN MODELS

| | System latency (ms) | | Lead time (ms) | |
|------------|---------------------|------------------|----------------------|----------------------|
| | CNN | Modified DAG-CNN | CNN | Modified DAG-CNN |
| Falls | 61.2 | 62.0 | 660.9 (150~1,575) | 662.0 (148~1,595) |
| Near-falls | 58.9 | 55.4 | - | - |
| ADLs | 56.4 | 58.2 | - | - |

To determine the superiority of the methodology used in this study, it was compared to previous methodologies related to binary classification for the prediction of near-fall (Suppl. Results and suppl. Table IV). By summarizing a few previous studies, machine learning- and threshold-based algorithms are generally used to predict near-fall [33]. The critical point in using threshold-based algorithms is to determine features to classify and their threshold values. However, it is out of the scope of this study to implement and determine the threshold feature for additional classification (falls vs. near-falls vs. ADLs) because most previous studies extract threshold features based on expert knowledge or experience (trial and errors). In addition, it is generally known that the machine learning model has higher performance than the classification by threshold-based model [48]. Therefore, previous methods using machine learning were compared with our method. They showed lower performances than our method. The decreased accuracy of the previous machine learning models might result from using a comprehensive dataset including 36 types of activities (10 types of falls, 10 types of near-falls, and 16 types of ADLs) instead of only a few types of activities. As mentioned in the previous paragraph, the heterogeneity characteristics of near-fall movements widely distributed between actual ADLs and fall regions might be a factor that could degrade classifier performance. Therefore, modified DAG-CNN architecture which expands the feature dimension plays a critical role in improving the performance of the classifier to precisely categorize falls, near-falls, and ADLs.

The ultimate goal of the project is to develop a protection system that monitors the risk of falls in advance and activates airbag devices if a fall is detected. The deep learning model proposed in this study will be used as the control algorithm of the protection system. In future work, it will be necessary to incorporate the proposed deep learning model into the control system. Optimization of the deep learning model taking into account the specifications and unit price of the control unit is necessary for system mounting. Additionally, the interface function should be implemented by linking with a smartphone application. Diagnosis of falls or near-falls should be transmitted through Bluetooth wireless communication to provide information that can be utilized by users or clinicians. If a problem occurs in Bluetooth communication (error in the transmitted data or disconnection out of range), an environment in which to reconnect by giving a notification (buzzer or vibration, etc.) to the user will be provided. Therefore, the fall, near-fall, and ADLs classification model proposed in this study has the potential to play a key role as the basic algorithm of the protection system.

The limitations of this study are as follows. First, a fall simulation experiment was conducted based on a scenario set in the laboratory, not an actual fall. The subjects were limited to young and healthy people because it could be very dangerous for elderly people with weakened balance or people with diseases such as dizziness to perform fall experiments. Nevertheless, since actual falls occur faster and with more force than in simulated experiments [49], the detection algorithm proposed in this study is sufficiently significant. Second, there is a limitation in not classifying the 36 types of activities (10 types of falls, 10 types of near-falls, and 16 types of ADLs) in detail. Detailed classification of the type or direction of the fall and near-fall can estimate the injury site, leading to the appropriate operation of the protection system [10]. In future research, additional in-depth study on detailed classification is necessary. Third, the fall detection algorithm of this study has a limitation in that only CNN-based deep learning models were used. Until now, various types of deep learning models have been developed, and it is known that the performance of the deep learning model varies depending on the characteristics of the training data [50]. However, this study proposed a comprehensive fall detection algorithm that can simultaneously predict falls, near-falls, and ADLs. Therefore, an in-depth study to compare and evaluate performance by applying various deep learning models within this process is needed in the future.

In conclusion, we have proposed a novel deep learning-based pre-impact fall detection algorithm that can precisely discriminate falls, near-falls, and ADLs using acceleration and angular velocity signals of a single IMU device. Custom-designed IMU device hardware was designed to obtain acceleration and angular velocity signals. The modified DAG-CNN structure was used to predict falls, near-falls, and ADLs, and a comparative evaluation with traditional CNN techniques was performed. As a result, the prediction results of the modified DAG-CNN structure were found to be approximately 7% more accurate than the traditional CNN structure. Additionally, the deep learning model trained by combining acceleration and angular velocity showed better performance than each model of acceleration and angular velocity. The results of this study will improve the performance of the fall detection system by reducing false alarms. It is believed that information to preemptively handle the risk of falls as well as quantitative evaluation of the rehabilitation status of the elderly with weak balance will be provided by monitoring of near-falls.

REFERENCES

- [1] A. Choi, T. G. Kang, and J. H. Mun, "Biomechanical evaluation of dynamic balance control ability during golf swing," *J. Med. Biol. Eng.*, vol. 36, no. 3, pp. 430–439, Jun. 2016.
- [2] A. Choi, T. Sim, and J. H. Mun, "Improved determination of dynamic balance using the centre of mass and centre of pressure inclination variables in a complete golf swing cycle," *J. Sports Sci.*, vol. 34, no. 10, pp. 906–914, Aug. 2015.
- [3] C. Hrysomallis, "Relationship between balance ability, training and sports injury risk," *Sports Med.*, vol. 37, no. 6, pp. 547–556, 2007.
- [4] A. Choi, E. Park, T. H. Kim, G. J. Im, and J. H. Mun, "A novel optimization-based convolution neural network to estimate the contribution of sensory inputs to postural stability during quiet standing," *IEEE J. Biomed. Health Inform.*, early access, Jun. 27, 2022, doi: 10.1109/JBHI.2022.3186436.

- [5] A. A. Zecevic, A. W. Salmoni, M. Speechley, and A. A. Vandervoort, "Defining a fall and reasons for falling: Comparisons among the views of seniors, health care providers, and the research literature," *Gerontologist*, vol. 46, no. 3, pp. 367–376, Jun. 2006.
- [6] F. De Backere *et al.*, "Towards a social and context-aware multi-sensor fall detection and risk assessment platform," *Comput. Biol. Med.*, vol. 64, pp. 307–320, Sep. 2015.
- [7] M. Á. Á. De La Concepción, L. M. S. Morillo, J. A. Á. García, and L. González-Abril, "Mobile activity recognition and fall detection system for elderly people using ameva algorithm," *Pervasive Mobile Comput.*, vol. 34, pp. 3–13, Jan. 2017.
- [8] M. V. Albert, K. Kording, M. Herrmann, and A. Jayaraman, "Fall classification by machine learning using mobile phones," *PLoS One*, vol. 7, no. 5, May 2012, Art. no. e36556.
- [9] J. K. Lee, S. N. Robinovitch, and E. J. Park, "Inertial sensing-based pre-impact detection of falls involving near-fall scenarios," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 23, no. 2, pp. 258–266, Mar. 2015.
- [10] T. H. Kim, A. Choi, H. M. Heo, K. Kim, K. Lee, and J. H. Mun, "Machine learning-based pre-impact fall detection model to discriminate various types of fall," *J. Biomech. Eng.*, vol. 141, no. 8, May 2019.
- [11] Y. Wu, Y. Su, R. Feng, N. Yu, and X. Zang, "Wearable-sensor-based pre-impact fall detection system with a hierarchical classifier," *Measurement*, vol. 140, pp. 283–292, Jul. 2019.
- [12] L.-J. Kau and C.-S. Chen, "A smart phone-based pocket fall accident detection, positioning, and rescue system," *IEEE J. Biomed. Health Inform.*, vol. 19, no. 1, pp. 44–56, Jan. 2015.
- [13] G. Shi, C. S. Chan, W. J. Li, K.-S. Leung, Y. Zou, and Y. Jin, "Mobile human airbag system for fall protection using MEMS sensors and embedded SVM classifier," *IEEE Sensors J.*, vol. 9, no. 5, pp. 495–503, May 2009.
- [14] M. N. Nyan, F. E. H. Tay, and E. Murugasu, "A wearable system for pre-impact fall detection," *J. Biomech.*, vol. 41, no. 16, pp. 3475–3481, 2008.
- [15] G. Wu and S. Xue, "Portable preimpact fall detector with inertial sensors," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 16, no. 2, pp. 178–183, Apr. 2008.
- [16] C. M. Arnold and R. A. Faulkner, "The history of falls and the association of the timed up and go test to falls and near-falls in older adults with hip osteoarthritis," *BMC Geriatrics*, vol. 7, no. 1, pp. 1–9, Jul. 2007.
- [17] I. Pang, Y. Okubo, D. Sturnieks, S. R. Lord, and M. A. Brodie, "Detection of near falls using wearable devices: A systematic review," *J. Geriatric Phys. Therapy*, vol. 42, no. 1, pp. 48–56, 2019.
- [18] S. N. Robinovitch *et al.*, "Video capture of the circumstances of falls in elderly people residing in long-term care: An observational study," *Lancet*, vol. 381, no. 9860, pp. 47–54, 2013.
- [19] N. H. Chehade, P. Ozisik, J. Gomez, F. Ramos, and G. Pottie, "Detecting stumbles with a single accelerometer," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2012, pp. 6681–6686.
- [20] B. Lindholm, P. Hagell, O. Hansson, and M. H. Nilsson, "Prediction of falls and/or near falls in people with mild parkinson's disease," *PLoS One*, vol. 10, no. 1, Jan. 2015, Art. no. e0117018.
- [21] A. M. Sabatini, G. Ligorio, A. Mannini, V. Genovese, and L. Pinna, "Prior-to- and post-impact fall detection using inertial and barometric altimeter measurements," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 24, no. 7, pp. 774–783, Jul. 2016.
- [22] M. Yu, Y. Yu, A. Rhuma, S. M. R. Naqvi, L. Wang, and J. A. Chambers, "An online one class support vector machine-based person-specific fall detection system for monitoring an elderly individual in a room environment," *IEEE J. Biomed. Health Inform.*, vol. 17, no. 6, pp. 1002–1014, Nov. 2013.
- [23] A. Choi, H. Jung, K. Y. Lee, S. Lee, and J. H. Mun, "Machine learning approach to predict center of pressure trajectories in a complete gait cycle: A feedforward neural network vs. LSTM network," *Med. Biol. Eng. Comput.*, vol. 57, no. 12, pp. 2693–2703, Nov. 2019.
- [24] N. E. El-Attar, M. K. Hassan, O. A. Alghamdi, and W. A. Awad, "Deep learning model for classification and bioactivity prediction of essential oil-producing plants from Egypt," *Sci. Rep.*, vol. 10, no. 1, pp. 1–10, Dec. 2020.
- [25] M. Nouredanesh, K. Gordt, M. Schwenk, and J. Tung, "Automated detection of multidirectional compensatory balance reactions: A step towards tracking naturally occurring near falls," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 2, pp. 478–487, Feb. 2020.
- [26] A. Weiss, I. Shimkin, N. Giladi, and J. M. Hausdorff, "Automated detection of near falls: Algorithm development and preliminary results," *BMC Res. Notes*, vol. 3, no. 1, p. 62, 2010.
- [27] J. M. H. Karel *et al.*, "Towards unobtrusive *in vivo* monitoring of patients prone to falling," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol.*, Aug. 2010, pp. 5018–5021.
- [28] T. Iluz *et al.*, "Automated detection of missteps during community ambulation in patients with Parkinson's disease: A new approach for quantifying fall risk in the community setting," *J. Neuroeng. Rehabil.*, vol. 11, no. 1, pp. 1–9, Apr. 2014.
- [29] A. R. Kammerdiner and A. N. Guerro, "Data-driven combinatorial optimization for sensor-based assessment of near falls," *Ann. Oper. Res.*, vol. 276, nos. 1–2, pp. 137–153, May 2019.
- [30] T. H. Kim, A. Choi, H. M. Heo, H. Kim, and J. H. Mun, "Acceleration magnitude at impact following loss of balance can be estimated using deep learning model," *Sensors*, vol. 20, no. 21, p. 6126, Oct. 2020.
- [31] Z. Zhong *et al.*, "A real-time pre-impact fall detection and protection system," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM)*, Jul. 2018, pp. 1039–1044.
- [32] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electron. Markets*, vol. 31, no. 3, pp. 685–695, Apr. 2021.
- [33] O. Aziz, M. Musngi, E. J. Park, G. Mori, and S. N. Robinovitch, "A comparison of accuracy of fall detection algorithms (threshold-based vs. machine learning) using waist-mounted tri-axial accelerometer signals from a comprehensive set of falls and non-fall trials," *Med. Biol. Eng. Comput.*, vol. 55, no. 1, pp. 45–55, Apr. 2016.
- [34] N. Pannurat, S. Thiemjarus, and E. Nantajeewarawat, "Automatic fall monitoring: A review," *Sensors*, vol. 14, no. 7, pp. 12900–12936, 2014.
- [35] L. Wang, M. Peng, and Q. Zhou, "Pre-impact fall detection based on multi-source CNN ensemble," *IEEE Sensors J.*, vol. 20, no. 10, pp. 5442–5451, May 2020.
- [36] J. O. P. Arenas, R. J. Moreno, and R. D. H. Beleno, "Convolutional neural network with a DAG architecture for control of a robotic arm by means of hand gestures," *Contemp. Eng. Sci.*, vol. 11, no. 12, pp. 547–557, 2018.
- [37] Y.-D. Zhang, C. Pan, J. Sun, and C. Tang, "Multiple sclerosis identification by convolutional neural network with dropout and parametric ReLU," *J. Comput. Sci.*, vol. 28, pp. 1–10, Sep. 2018.
- [38] B. Kim, N. Yuvaraj, K. R. S. Preethaa, and R. A. Pandian, "Surface crack detection using deep learning with shallow CNN architecture for enhanced computation," *Neural Comput. Appl.*, vol. 33, no. 15, pp. 9289–9305, Jan. 2021.
- [39] W. Jia, C. Xiu-Yun, Z. Hao, X. Li-Dong, L. Hang, and D. Si-Hao, "Hyperparameter optimization for machine learning models based on Bayesian optimization," *J. Electron. Sci. Technol.*, vol. 17, no. 1, pp. 26–40, 2019.
- [40] F. Mun and A. Choi, "Deep learning approach to estimate foot pressure distribution in walking with application for a cost-effective insole system," *J. Neuroeng. Rehabil.*, vol. 19, no. 1, pp. 1–14, Jan. 2022.
- [41] T. Tamura, T. Yoshimura, M. Sekine, M. Uchida, and O. Tanaka, "A wearable airbag to prevent fall injuries," *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, no. 6, pp. 910–914, Nov. 2009.
- [42] E. Casilari, J.-A. Santoyo-Ramón, and J.-M. Cano-García, "Analysis of public datasets for wearable fall detection systems," *Sensors*, vol. 17, no. 7, p. 1513, Jun. 2017.
- [43] M. Trkov, K. Chen, J. Yi, and T. Liu, "Inertial sensor-based slip detection in human walking," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 3, pp. 1399–1411, Jul. 2019.
- [44] E. A. Shammam, A. T. Zahary, and A. A. Al-Shargabi, "A survey of IoT and blockchain integration: Security perspective," *IEEE Access*, vol. 9, pp. 156114–156150, 2021.
- [45] S. Yang and D. Ramanan, "Multi-scale recognition with DAG-CNNs," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1215–1223.
- [46] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.
- [47] S. Taheri and Ö. Toygar, "On the use of DAG-CNN architecture for age estimation with multi-stage features fusion," *Neurocomputing*, vol. 329, pp. 300–310, Feb. 2019.
- [48] S. Rastogi and J. Singh, "A systematic review on machine learning for fall detection system," *Comput. Intell.*, vol. 37, no. 2, pp. 951–974, Apr. 2021.
- [49] J. M. Anglin, T. Sugiyama, and S.-L. Liew, "Visuomotor adaptation in head-mounted virtual reality versus conventional training," *Sci. Rep.*, vol. 7, no. 1, p. 45469, Apr. 2017.
- [50] J. S. Sánchez, R. A. Mollineda, and J. M. Sotoca, "An analysis of how training data complexity affects the nearest neighbor classifiers," *Pattern Anal. Appl.*, vol. 10, no. 3, pp. 189–201, Aug. 2007.