PNF and Movement, 2024; 22(1): 43-54
https://doi.org/10.21598/JKPNFA.2024.22.1.43

Print ISSN: 2508-6227
Online ISSN: 2508-6472

Original Article

Open Access

Vest-type System on Machine Learning-based Algorithm to Detect and Predict Falls

Ho-Chul Kim, Ph.D.[1,2,3] • Ho-Seong Hwang, M.S.[2,3] •
Kwon-Hee Lee, M.S.[4] • Min-Hee Kim, P.T., Ph.D.[5†]

[1]Department of Radiological Science, College of Health Science, Eulji University
[2]Research Institute of Machine Intelligence Convergence System, EulJi University
[3]Department of Medical Artificial Intelligent, Graduate school of Eulji University
[4]Department of Bio-Medical Science, Korea University
[5]Department of Physical therapy, College of Health Science, Eulji University

Received: March 6, 2024 / Revised: March 21, 2024 / Accepted: March 22, 2024

| Abstract |

Purpose: Falls among persons older than 65 years are a significant concern due to their frequency and severity. This study aimed to develop a vest-type embedded artificial intelligence (AI) system capable of detecting and predicting falls in various scenarios.
Methods: In this study, we established and developed a vest-type embedded AI system to judge and predict falls in various directions and situations. To train the AI, we collected data using acceleration and gyroscope values from a six-axis sensor attached to the seventh cervical and the second sacral vertebrae of the user, considering accurate motion analysis of the human body. The model was constructed using a neural network-based AI prediction algorithm to anticipate the direction of falls using the collected pedestrian data.
Results: We focused on developing a lightweight and efficient fall prediction model for integration into an embedded AI algorithm system, ensuring real-time network optimization. Our results showed that the accuracy of fall occurrence and direction prediction using the trained fall prediction model was 89.0% and 78.8%, respectively. Furthermore, the fall occurrence and direction prediction accuracy of the model quantized for embedded porting was 87.0 % and 75.5 %, respectively.
Conclusion: The developed fall detection and prediction system, designed as a vest-type with an embedded AI algorithm, offers the potential to provide real-time feedback to pedestrians in clinical settings and proactively prepare for accidents.

Key Words: Accidental Falls, Neural network, Safety Management, Risk Assessment

†Corresponding Author : Min-Hee Kim (kmh12@eulji.ac.kr)

Ⅰ. Introduction

Falls are important causes of fatal and common injuries in persons older than 65 years. Annually, 2.8 million people visit an emergency center for a fall, and 25% of falls suffer serious injuries including fractures or brain injury (Moncada & Mire, 2017). In particular, the importance of establishing falls and fall-related injuries prediction system is increasing because of the large population of older people living alone. For these reasons, fall prediction system continues to grow in importance (Ajerla et al., 2019; Angal& Jagtap, 2016; Ren & Peng, 2019; Xu et al., 2018).

Studies for fall prediction have been conducted using computer vision methods with cameras and acceleration sensors, but some of these have several limitations and disadvantages (Cucchiara et al., 2007). In previous studies using computer vision and cameras based on image data sets, the body part of subjects could be obscured by an unspecified object in single camera and appeared difficulties of not being able to detect falls. For solving these problems, the method using multiple cameras such as Kinect and depth-based three-dimensional (3-D) multi-camera has been used predominantly. However, these methods are costly and computationally heavy. Moreover, special-purpose cameras, such as Kinect and 3-D cameras, have distance constraints, which impose practical and real-life applicability difficulties.

Another way is to use sensors of acceleration and angular velocity, values were obtained with the necessary equipment attached to the subject. This paradigm has been used to formulate relevant equations to determine fall occurrence (Huynh et al., 2015; Kwolek & Kepski, 2015). However, owing to the various physical conditions that must be accounted for, such as user's height and weight, it is not appropriate to determine a fall by merely substituting the parameters in the formalized equation. To address these difficulties, research on fall determination algorithms based on artificial intelligence (AI) is actively drawing interest. These complex methods using AI have shown satisfactory performance in predicting falls in preliminary studies (Kerdegari et al., 2013; Shi et al., 2020). However, methods with AI required high-speed computing devices and were expensive, so that were not suitable for wearable devices. Additionally, another approach was to apply a communication-based system, but this could not provide immediate feedback to the pedestrian when communication status problems occur (Musci et al., 2020).

Therefore, we recognized the need to develop the new approach to predict the fall and fall direction based on neural networks andan optimization method, which in turn is based on a model executed in an embedded real-time clinic environment (Mascret et al., 2018). Therefore, we design a new system based on embedded-AI fall prediction and detection and prediction model and versatile vest-type system that protects falls for older adults. This approach would be effective to detect and predictfalls to some extent by transplanting a lightweight prediction model into a vest-type system to process pedestrian data in real-time.

Ⅱ. Methods

1. Experiment Participants

This study was conducted with prior approval from the Institutional Review Board (IRB) of Eulji University (approval number: EU18-36, date of examination: 10/25/2019).Participants were selected without any problems during walking. The experiment was performed after participants signed the consent form to participate in the experiment. A total of 80 people participated in the experiment, including 30 seniors aged 65 to 79 (71.70

± 5.72,7 men and 23 women) and 50 adults aged 20 to 29 (21.75±1.44,30 men and 20 women).The senior group had height of 158.08±5.88cm, weight of 57.50 ±6.91kg, and BMI of 23.01±2.56kg/m^2.The adult group had height of 169.92±8.63cm, weight of 66.68 ±6.97kg, and BMI of 23.31±2.99kg/m^2.

2. Falls data collecting equipment

1) Falls data collecting equipment

The protective vest was selected as a basically protective, inexpensive, and practical product. The vest is as shown in Figure 1. It was used for detecting falls the 6-axis sensor acceleration and gyroscope (as gyro) output along the x, y, and z axes. Acceleration and gyro values accurately simulated pedestrian movements and changes that occur during falls. The three axes of acceleration each represented the degree of tilt and rotation based on the direction of gravity. Acceleration and gyro values that were the same in each axis direction had positive values, and those in the opposite direction had negative values. The selected sensor was small that it could be easily attached to participants 13. Additionally, each sensor had low power consumption and was easy to use, making it suitable for building embedded systems. The sensor was attached to the vest and received acceleration and gyro sensor data. In detail, sensors were installed at the 7th cervical vertebrae and the 2nd sacrum of each participant, because the lateral movement of the body by walking could be minimized in Figure 1(a). At the same time, sensor values by human movement depending on height were acquired with variety, enabling more accurate motion analysis.

2) Data processing embedded system

The main embedded module consisted of a microcontroller, IMU (Inertial Measurement Unit), Bluetooth and SD card slot. This main module was a

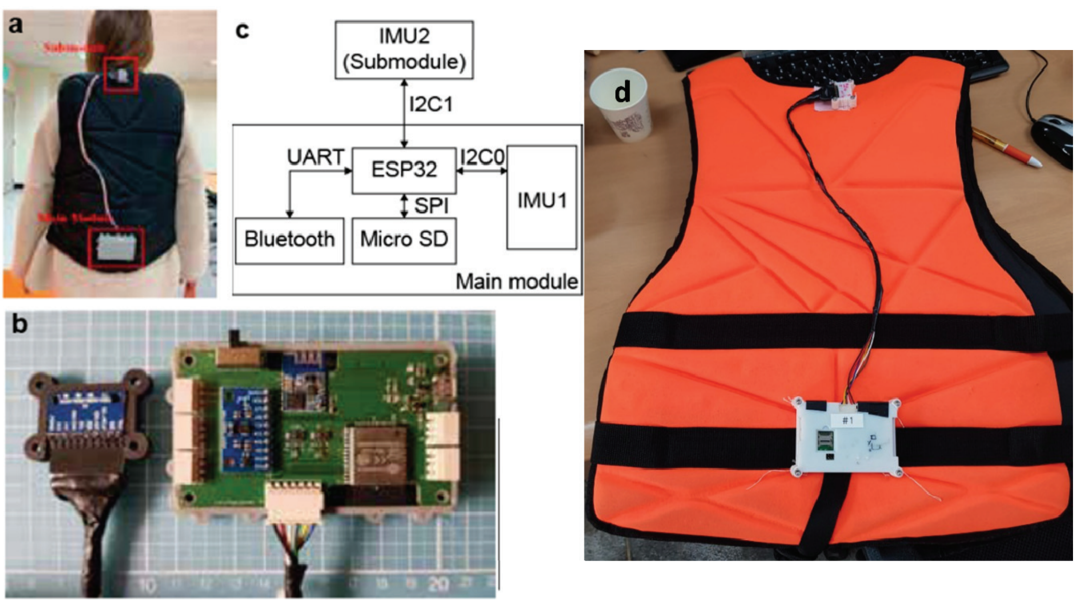

Fig. 1. (a) Sensor attachment positions, (b) Submodule and main module, (c) Block diagram of the system, (d) Detailed protective vest for falling test.

data logger that collects training data and a prediction system(Babiuch et al., 2019). The submodule only contained the IMU, only performed the role of acquiring data from sensors and transmitting data to the main module (Figure 1).

3. Falls data acquisition settings

1) Fall patterns

The sensor data were collected from the participants and represent time series data - an appropriate number of data samples were required to infer the tendencies of the patterns. To train the fall detection algorithm, it must be obtained the acceleration and gyro values from a normal walking pattern (gait pattern). After then, data during and after a fall were collected. However, fall experiments were not conducted with the participants older than 30 years to prevent safety accidents. Participant performed falling, normal walking, and running movements in different directions (forward, backward, left, and right) five times each.Specifically, normal walking and running each covered 6 meters distance, during which the participant walked for 6 seconds and ran for 3 seconds, as shown in Figure 2(a) and (b).

The main module and submodule data obtained during

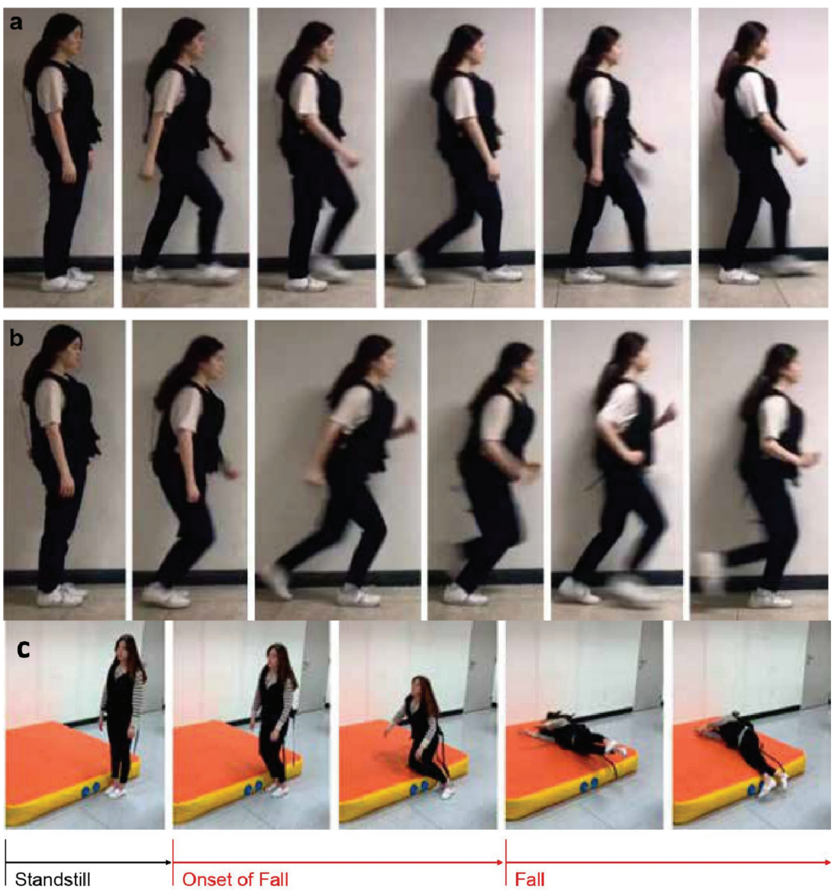

Fig. 2. Image montage at various timeframes showing a pedestrian (a) walking and (b) running and participant's (c) falling to the right.

normal walking and running with the passage of time. During falls, participants fell in four directions (front, back, left, right) while being stopped. Acceleration and gyro data at this time were acquired and used.Based on the data, falls were classified into three patterns, as shown in Figure 2(c), which was an example of a participant falling to the right. Figure 3 shows the changes in the acceleration (a) and gyro (b) values of the main module according to each situation during falls.

2) Data acquisition time

The enlargement of the main module acceleration and gyro values shown in Figure 3(a) and (b), respectively.The

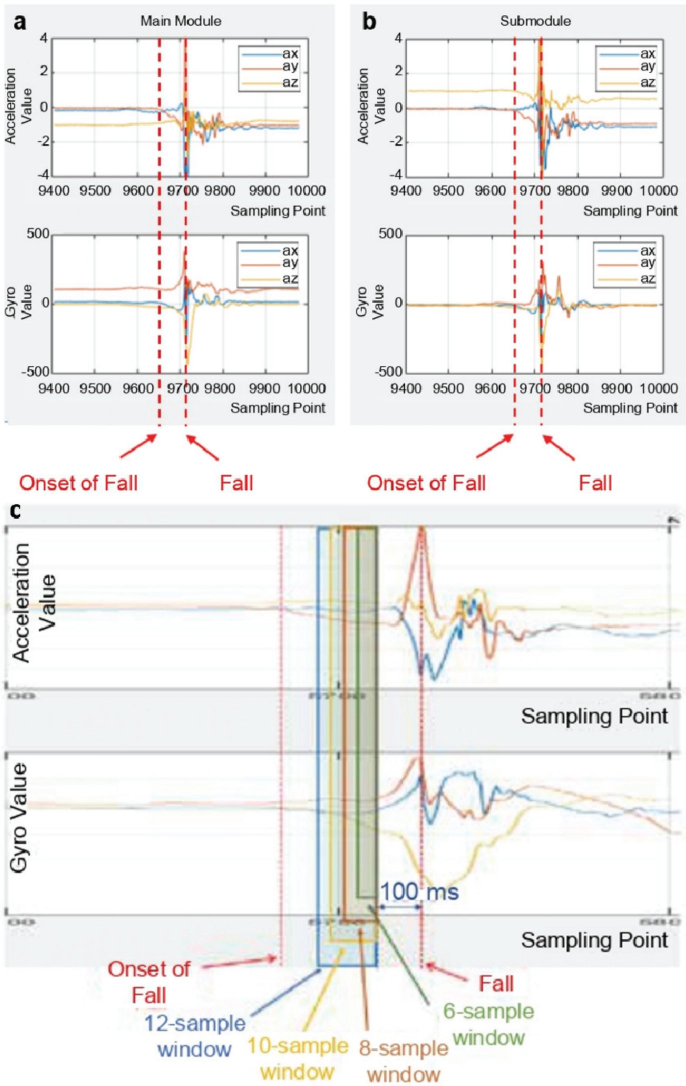

Fig. 3. Sensor pattern during falls to the right for the (a) main module and (b) submodule and (c) data sample window participants required for fall prediction.

acceleration and gyro value data from the main and sub-sensors were collected as theparticipant began to fall to the right from a standing position. The large change in the gyro value begins with the actual fall (Wang et al., 2017). From this, it was identified that the time from the beginning to the end of the fall was about 600ms. Thus, it was set that acceleration and gyro values were collected every 10ms from the sensor module attached to the participants.

4. Data sampling

It is important to determine the appropriate number of samples to train the algorithmic model by reproducing the sensor values generated when a pedestrian falls in the form of a continuous signal.If the number of samples is too small, it is difficult to identify the movement without obtaining enoughfeatures explaining the fall of the participant. If the number of samples is too large, the amount of data that must be processed increases, resulting in increased computational load and over-fitting of features.

As a result of analyzing the movement of participants in the experiment, the walking and running frequencies were found to be about 2Hz and 3Hz, respectively. Therefore, when acquiring acceleration and gyro data, samples should be taken at a minimum frequency of 6 Hz according to the movement to avoid aliasing (Cook 1986). However, when sampling at 6Hz, it is expected that it will be difficult to accurately measure smallmovements during falls.Therefore, we oversampled it at a frequency of 100Hz to measure participant movement more accurately.

As shown in Figure 4(a), if the number of samples was 3 and a fall occurs at the time of falling, the model should use the sampling data S2, S1, and S0.As a result of the analysis of participants' data showed that the time from onset to fall within 600ms was approximately 420-680ms.

Based on this, we used data from the sample window area 100ms before the fall to determine the participant group, as shown by the blue stripin Figure 3(c). Considering that the data sampling frequency of the

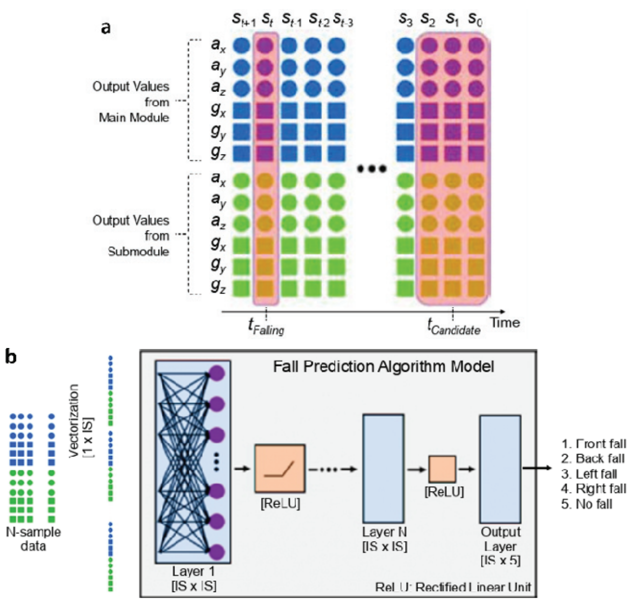

Fig. 4. (a) Sensor output and training set participants over time, (b) Model structure of the fall prediction algorithm.

sensors attached to the participants was 100Hz, the computation time did not exceed 10ms for the algorithm to operate in real time. The ESP32 was a 240MHz microcontroller. It then computed a fall prediction algorithm. ESP32requiredapproximately 0.054µs to perform floating point multiplication and addition. Therefore, up to 12 sample sets could be used. Therefore, we used four fall prediction candidate datasets from 6, 8, 10, and 12 sample sets for AI learning. Finally, 366 data sets were used, of which 256 (70%) were used as training sets and the remaining 110 (30%) were used as test sets.

5. Neural network-based fall prediction algorithm

Figure 4(b) shows thestructure of the fall prediction algorithm used in this study. As a set of up to 12 samples was used, 12 input data points (I) were used, consisting of acceleration and gyro values for each of the three axes of the main module and submodule. Multiplying this by the number of samples (S) results in the total number of data points input to the fall prediction algorithm is calculated as shown in Equation 1. For example, if movement data spanning 100 mswas used as input, 120 input data (12 × 10 samples) are used for prediction.

$$\text{Number of Input Data Points} = \text{Number of Sensor Ouputs (I)} \cdot \text{Number of Samples (S)}$$

The proposed prediction algorithm consists of five output classes, where 'I' represent the 12 sensor values collected in the experiment. The input data matrix [1 × IS] passes through the hidden layer matrix [IS × IS], and a rectifier linear unit (ReLU) was used as the activation function for the nonlinearity of the algorithm. Finally, it passed through the N-hidden layer and was classified into five classes because there were four different fall directions and one case where no fall occurs. The fall prediction algorithm usedthe output values from the main module and submodule to predict whether a fall will occur, and if a fall is predicted, it also predicts the direction of the fall. The cross-entropy cost function was used to measure the discrepancy between predicted and actual values, and the Adam optimizer was used to minimize the cost function. During the training process, 100 epochs were run, and dropout normalization was applied to 80-90% of the neurons to mitigate overfitting.

6. Weight quantization

Approximately 6.8ms and 9ms were required for three and four hidden layers, respectively. Considering the calculations omitted from the computational prediction and the acceleration of the 6-axis sensor of the ESP32 microcontroller and the delay in the gyro value, at least three hidden layers should be selected to ensure real-time performance of the algorithm. Due to the lack of hidden

Table 1. Optimal architectural dimensions

Layer	Shape	Weights	Weight Size (FP32)	Weight Size (Quantized)
Hidden 1 + rectified linear unit (ReLU)	120 × 120 + 120	14,520	58,080	14,520
Hidden 2 + ReLU	120 × 120 + 120	14,520	58,080	14,520
Hidden 3 + ReLU	120 × 120 + 120	14,520	58,080	14,520
Hidden 4 + ReLU	120 × 120 + 120	14,520	58,080	14,520
Output	120 × 5 + 5	605	2,420	605
Total		**58,685**	**234,740 Bytes**	**58,685 Bytes**

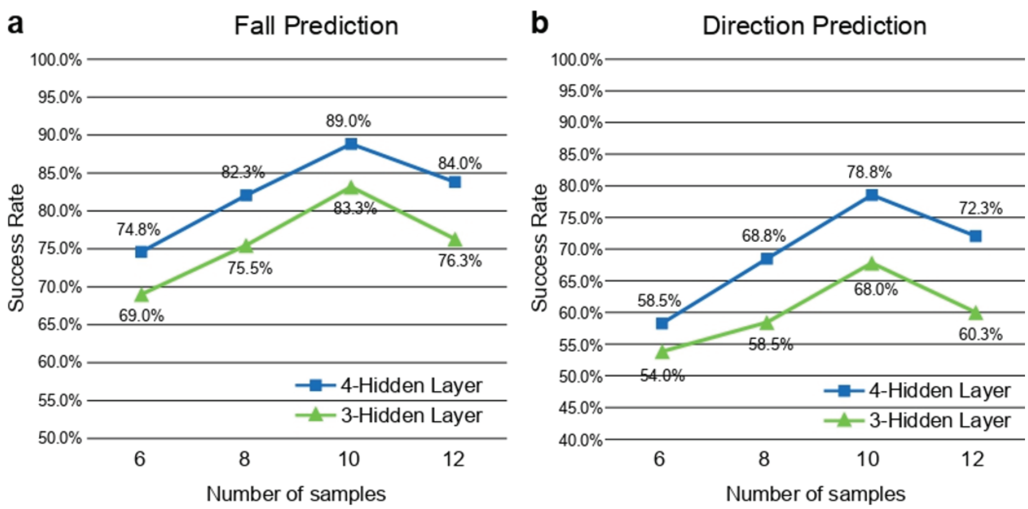

Fig. 5. (a) Success rate of fall prediction as a function of the number of layers. (b) Success rate of fall direction prediction as a function of the number of layers.

layers, the desired performance may not be achieved, but increasing the number of layers may prevent real-time performance from being achieved. Figure 5 shows the accuracy according to the composition of 3-and 4-hidden layer. The predictive algorithm model consisted of 10 samples and four hidden layers, with 58,685 learning weights. Table 1 shows the algorithm architectures ported to embedded systems, weight sizes for learning, and data sizes.

7. Weight quantization

Once the number of input data in the sample and the number of layers in the model had been determined, the model was trained using a training set. Since the storage of a weight in memory requires at least 4 bytes of space, the trained model requires 234,740 bytes of memory space for the storage the total number of weights. However, this does not use external dynamic random access memory. It is not suitable for the ESP32 hardware used in this study and significantly reduces the computational speed for hardware that does not use a real accelerator.

We converted from 32-bit floating point to 8-bit integer to reduce the size of the weight used in the artificial intelligence algorithm. Accordingly, the resulting change in performance was compared and evaluated. 32-bit floating point numbers arereferred to as FP32, and 8-bit integers are referred to as INT8.

The following 120 × 120 $W^{[1]}_{120\times120}$matrix contains the trained weights from the first layer, which are quantized to INT8 and are as follows:

$$W^{[1]}_{120\times120} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,120} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,120} \\ \vdots & \vdots & \ddots & \vdots \\ w_{120,1} & w_{120,2} & \cdots & w_{120,120} \end{bmatrix} = \begin{bmatrix} -0.2550 & -0.6861 & \cdots & -0.6917 \\ 0.4159 & -0.0325 & \cdots & -0.0326 \\ \vdots & \vdots & \ddots & \vdots \\ 0.5423 & -0.7424 & \cdots & 0.9418 \end{bmatrix} \quad (2)$$

The maximum and minimum values of the matrix are listed in Table 2. INT8 can represent 256 integers between -128 and 127. Therefore, the maximum value of the W matrix element, 1.0193, corresponds to 127, the minimum value -1.0268 corresponds to -128, and the values between the maximum and minimum values could be expressed in 256 steps. Accordingly, we quantized the weights of the 3- and 4- hidden layer models and evaluated the final performance using the test set.

Ⅲ. Results

The performance of the fall prediction algorithm model was evaluated using a test set. The results of the quantized fall prediction model are shown in Figure 6. In Figure 6, the green line represents the prediction success rate according to the number of samples when the algorithm consists of three hidden layers.Figure 6(a) shows the fall prediction results regardless of the direction at the time of the fall, and Figure 6(b) shows the prediction accuracy results for not only for the occurrence of the fall but also for the direction of the fall. As a result, the appropriate setting for the fall prediction algorithm model is a model consisting of 10 unquantized samples and 4 hidden layers, with 58,685 learning weights.

Ⅳ. Discussion

The fall can be an unpredictable event and limitthe

Table 2. Minimum and maximum values after training

Layer	Shape	Minimum Value	Maximum Value
W1, B1	120 × 120, 1 × 120	−1.0268	1.0193
W2, B2	120 × 120, 1 × 120	−1.0176	1.0245
W3, B3	120 × 120, 1 × 120	−1.0209	1.0174
W4, B4	120 × 120, 1 × 120	−1.0202	1.0230
W5, B5	120 × 5, 1 × 5	−1.0039	1.0226

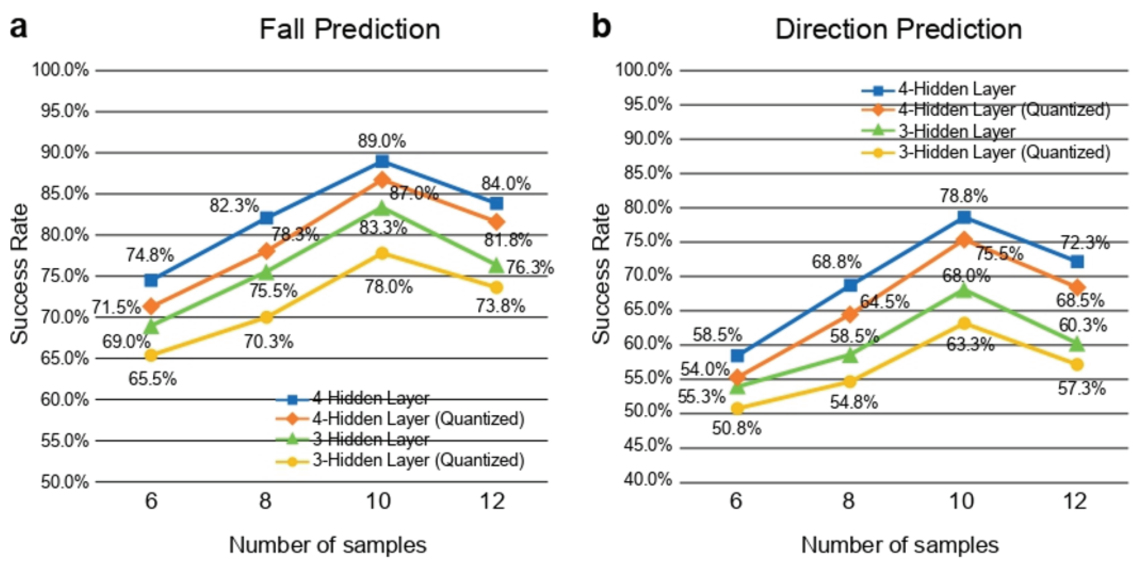

Fig. 6. (a) Success rate of fall prediction before and after quantization. (b) Success rate of fall direction prediction before and after quantization.

activities of daily living due to fear in the older adults. Therefore, the development of wearable device system for tracking, detecting, and preventing a fall for the older adults is very important to improve of the quality of life. While many previous studies have developed sensors and algorithms to predict and detect falls, they have lacked usability and practicality (Usmani et al., 2021; Zheng et al., 2022).

We assess falls based on embedded portable neural networks and design the algorithm models for prediction accordingly. Unlike previous fall determination methods, this method can predict falls to provide feedback to pedestrians and prepare for accidents in advance. In order to design a fall prediction model that can be computed in real-time, we present a method for selecting a neural network model size. In addition, to apply the model to embedded systems with limited hardware resources, the model was lightened based on quantization. Moreover, this study was conducted to improve the generalizability of the fall prediction algorithm model by sampling without separating the senior and adult data. This approach can use the flexibility of AI learning methods to explore a wide range of applicability, even in situations with limited data, opening up a wider range of applications.

The fall prediction algorithm model trained based on the method used in this study had 89.0% and 78.8% accuracy in predicting falls and fall directions, respectively. In addition, the quantized models for embedded porting had 87.0% and 75.5% accuracy in predicting falls and fall directions, respectively.

As a result, the number of input samples of the model and the optimal number of hidden layers were derived experimentally. In Figure 6, the green line shows the predicted success rate according to the number of samples when the algorithm is composed of three hidden layers. Because the data collected from pedestrians is time series data, an appropriate number of samples is required to infer pattern trends. The experimental results demonstrate that the 10 samples have the highest success rate in predicting fall and fall direction. The blue line shows the prediction success rate according to the number of samples when four hidden layers were used. The prediction success rate was slightly higher than when using the 3-hidden layer configuration. As the number of training variables increases with each additional layer, so does the degree of freedom of the neural networks and the success rate of fall prediction. However, because computational load and training variables increase with the number of hidden layers, optimization of embedded systems with limited hardware resources may be disadvantageous.

Quantization has been used to create a lightweight model that can operate efficiently even in embedded systems with limited hardware resources. Although the model was lightweight based on the reduction in weight size due to quantization, it was found that the success rate between the two models was reduced by approximately 2-4% due to the influence of quantization errors. Because the maximum and minimum value ranges of the FP32 weight are quantized to 256 steps, the resolution that can be achieved at each step is not sufficient. In particular, the closer the weight value is to zero before quantization, the greater the quantization error. Therefore, when using quantization for weight reduction, the learning weight should be prevented from converging to zero during model learning to minimize error.

Mounting sensor modules on the back of pedestrians is cumbersome, however, can be useful compared to the limitations of image-based algorithms when used to facilitate fall prediction models on smartphones. In addition, the proposed model can be applied to relatively inexpensive hardware and can predict falls and fall directions that can be prevented by applying it to a pedestrian protection system.

The limitation of this study was that the experimental framework designed does not include falls that occur in completely unpredictable situations. In this study, scenarios were created to represent falls that are typically likely to occur, and then participants were asked to enact these falls. Future studies should present research methods that completely overcome these limitations.

The development of AI algorithms for falls could revolutionize the field of physical therapy. The development of AI algorithms for fall prediction can be used to detect and predict the risk of falls at an early stage by analyzing the gait patterns, balance, motor function, and the others in various patients. It can also help to provide personalized physical therapy by considering the patient's fall risk factors. These implications are expected to improve patient outcomes in physical therapy and contribute to the development of the field of physical therapy.

Ⅴ. Conclusion

The fall prediction model proposed in this study can be made more robust by learning with additional data. In addition, quantization learning should be applied so that the model is quantized during the learning process to minimize performance degradation due to quantization errors caused by efforts to make the modellightweight.

Acknowledgments

This research was funded by the National Research Foundation of Korea (NRF) grant funded by the Eulji government Ministry of Science, ICT & Future Planning, grant number 2016R1C1B2012888. The funding source had no role in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

Ajerla D, Mahfuz S, Zulkernine F. A real-time patient monitoring framework for fall detection. *Wireless Communications and Mobile Computing*. 2019;1-13.

Angal Y, Jagtap A. Fall detection system for older adults. *Institute of Electrical and Electronics Engineers Conference.* 2016;262-266.

Babiuch M, Foltýnek P, Smutný P. Using the ESP32 microcontroller for data processing. *Institute of Electrical and Electronics Engineers Conference.* 2019;1-6.

Cook RL. Stochastic sampling in computer graphics. *ACM Transactions on Graphics*. 1986;5(1):51-72.

Cucchiara R, Prati A, Vezzani R. A multi-camera vision system for fall detection and alarm generation. *Expert Systems*. 2007;24(5):334-345.

Gupta HP, Chudgar HS, Mukherjee S, et al. A Continuous Hand Gestures Recognition Technique for Human-Machine Interaction Using Accelerometer and Gyroscope Sensors. *IEEE Sensors Journal*. 2016;16:6425-6432.

Huynh QT, Nguyen UD, Irazabal LB, et al. Optimization of an accelerometer and gyroscope-based fall detection algorithm. *Journal of Sensors*. 2015;1-8.

Kerdegari H, Samsudin K, Rahman Ramli A, et al. Development of wearable human fall detection system using multilayer perceptron neural network. *International Journal of Computational Intelligence Systems*. 2013;6(1):127-136.

Kwolek B, Kepski M. Improving fall detection by the use of depth sensor and accelerometer. *Neurocomputing*. 2015;168:637-645.

Mascret Q, Bielmann M, Fall C-L, et al. Real-time human physical activity recognition with low latency prediction feedback using raw IMU data. *Institute of Electrical and Electronics Engineers Conference.* 2018;239-242.

Moncada LVV, & Mire LG. Preventing Falls in Older Persons. *American Family Physician*. 2017;96(4):240-247.

Musci M, De Martini D, Blago N, et al. Online fall detection using recurrent neural networks on smart wearable devices. *IEEE Transactions on Emerging Topics in Computing*. 2020;9(3):1276-1289.

Ren L, Peng Y. Research of Fall Detection and Fall Prevention Technologies: A Systematic Review. *IEEE Access*. 2019;7:77702-77722.

Shi J, Chen D, Wang M. Pre-impact fall detection with CNN-based class activation mapping method. *Sensors*. 2020;20(17):4750.

Usmani S, Saboor A, Haris M, et al. Latest Research Trends in Fall Detection and Prevention Using Machine Learning: A Systematic Review. *Sensors (Basel)*. 2021;21(15):5134.

Wang Y, Wu K, Ni LM. WiFall: Device-free fall detection by wireless networks. *IEEE Transactions on Mobile Computing.* 2016(2);581-594.

Xu T, Zhou Y, Zhu J. New Advances and Challenges of Fall Detection Systems: A Survey. *Applied Sciences*. 2018;8(3):418.

Zheng L, Zhao J, Dong F, et al. Fall Detection Algorithm Based on Inertial Sensor and Hierarchical Decision. *Sensors (Basel)*. 2022;23(1):107.