

Article

Fall Recognition System to Determine the Point of No Return in Real-Time

Bae Sun Kim, Yong Ki Son, Joonyoung Jung, Dong-Woo Lee and Hyung Cheol Shin *

Human Enhancement & Assistive Technology Research Section, Electronics and Telecommunications Research Institute, 218 Gajeong-ro, Yuseong-gu, Daejeon 34129, Korea; bskim72@etri.re.kr (B.S.K.); handcourage@etri.re.kr (Y.K.S.); joonyoung@etri.re.kr (J.J.); hermes@etri.re.kr (D.-W.L.)

* Correspondence: shin@etri.re.kr

Abstract: In this study, we collected data on human falls, occurring in four directions while walking or standing, and developed a fall recognition system based on the center of mass (COM). Fall data were collected from a lower-body motion data acquisition device comprising five inertial measurement unit sensors driven at 100 Hz and labeled based on the COM-norm. The data were learned to classify which stage of the fall a particular instance belongs to. It was confirmed that both the representative convolutional neural network learning model and the long short-term memory learning model were performed within a time of 10 ms on the embedded platform (Jetson TX2) and the recognition rate exceeded 94%. Accordingly, it is possible to verify the progress of the fall during the unbalanced and falling steps, which are classified by subdividing the critical step in which the real-time fall proceeds with the output of the fall recognition model every 10 ms. In addition, it was confirmed that a real-time fall can be judged by specifying the point of no return (PONR) near the point of entry of the falling down stage.

Keywords: fall detection; inertial measurement unit sensor; point of no return



Citation: Kim, B.S.; Son, Y.K.; Jung, J.; Lee, D.-W.; Shin, H.C. Fall Recognition System to Determine the Point of No Return in Real-Time. *Appl. Sci.* **2021**, *11*, 8626. <https://doi.org/10.3390/app11188626>

Academic Editor: Federico Divina

Received: 5 August 2021

Accepted: 13 September 2021

Published: 16 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Falls have long been researched from various perspectives [1–5]. Falls among the elderly have emerged as a social issue owing to an exponential increase in the elderly population [6]. A survey on the welfare and living conditions of the elderly (2017), published by the Ministry of Health and Welfare (MOHW), showed that the fall rate of seniors was 15.9% in 2017 [6].

Existing studies on falls can be mainly classified according to whether the fall detection systems used are based on camera images or inertial measurement units (IMUs). When protective devices for preventing falls—such as airbags—are used, fall detection systems based on IMUs are generally utilized. Previous studies on fall detection systems that apply IMUs focused on calculating a threshold based on the significant impacts generated by falls or a rapid change in acceleration or developing measures or services that can instantly cope with falls [7–9]. With the development of learning-based algorithms since 2010, research [10–12] on utilizing fall data obtained based on computer vision and data obtained by IMU sensors [13–16] has been actively conducted. To utilize a learning-based algorithm, a data set containing the characteristics of the problem to be solved is required; more datasets provide better results. The sisFall dataset, which reflects 15 types of falls based on five cases for each fall type, was recently released [17]. However, the volume of fall data in this dataset was insufficient.

In some studies, falls were determined by considering areas related to the center of mass (COM) and the base of support (BOS) of human beings as feature points based on data collected by Kinect cameras [18,19]. Xu Tao (2017) predicted falls based on data obtained by Kinect cameras and a long short-term memory (LSTM) algorithm. Specifically, falls were predicted under the condition of 5 and 10 frames before the standard moment

of collision with the floor. However, the prediction result (75%) obtained when applying the condition of five frames before the standard moment was less accurate than under the condition of 10 frames (91.7%) [19].

Similar to the public sisFall data set, the fall data set collected in existing fall studies was collected by designing and mimicking the fall situation in several cases and the data set quantity was insufficient. To reflect the characteristics of falls that occur in real situations, a method is needed for reproducing a realistic fall situation. In addition, a method for collecting a large amount of data and for determining the progress of the fall in real-time is needed.

Thus, this study proposes a real-time fall progress detection system that can reflect practical falls that occur in daily life, e.g., when walking or standing. Hence, fall data were obtained via a fall implementation device that can reproduce realistic fall situations to reduce artificial elements that might be generated in a laboratory and maximize the volume of fall data obtained. Moreover, a learning model was operated based on a unit of 10 ms to monitor the progress of falls and ultimately determine falls. In addition, IMU sensor data were utilized for wearable situations. We thus developed a lower limb motion capture device, visualization S/W monitoring motion data, and a fall-reproducing device that induces falls. These devices were utilized to collect fall data and the fall data collection experiment was conducted with the approval of the Institutional Review Board (IRB) at Konyang University (project no.: KYU-2020-055-02).

The remainder of this paper is organized as follows. In Section 2, the critical phase of falls is divided into an imbalance phase and a falling phase to facilitate real-time fall recognition, and COM-based labels are described. In Section 3, the devices and methods used for data collection are described. The learning and recognition results are presented and analyzed in Section 4. In Section 5, the real-time implementation results based on the target platform and verification method are presented. Finally, conclusions and future research directions are presented in Section 6.

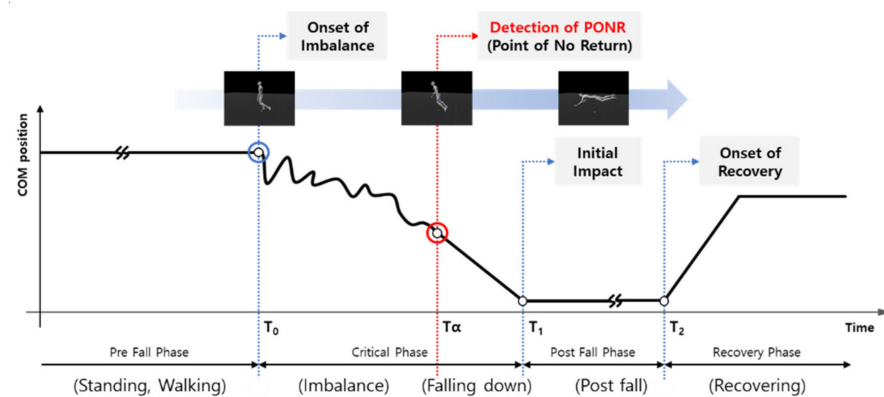
2. Materials and Methods

2.1. Sub-Division of the Critical Phase

A fall, defined in an existing study, is a sequential process of a person losing balance during his or her daily activities, colliding with the ground, lying on the ground, and standing up again. In this study, the entire phases of falling were divided into pre-fall, critical, post-fall, and recovery phases [3]. This study aimed to detect the critical phase, as presented in Table 1, i.e., the point where a person cannot recover his or her position during the fall process. The pre-fall phase, which refers to daily activities, was limited to falls generated by standing and walking. The critical phase, which refers to a fall progress process, was sub-divided into two specific phases, as shown in Figure 1. In this figure, the x- and y-axes indicate the time flow and location of the COM in the direction of gravity, respectively. T_0 indicates the moment when balance is disrupted by a causal factor, leading to a fall, and T_0 to T_α is the period during which a person can recover balance, according to his or her exercise ability. The T_α point indicates when the imbalance has progressed to some extent; after this, it is difficult to recover the posture using exercise ability. This study aimed to detect the point of no return (PONR), i.e., when a person cannot recover his or her balance based on his or her exercise ability. T_1 refers to the point at which the chest or bottom of a person touches the ground. The time after this point indicates a complete fall.

Table 1. Sub-division of the critical phase among the entire phases of the fall progress.

Fall Phases	Sub-Division of the Critical Phase	Classification of Behaviors
Pre-fall phase	Standing (S) Walking (W)	Daily activities
Critical phase	Imbalance (I) Falling down (F)	Fall in progress
Post fall phase	Post Fall (P)	After falling down
Recovery phase	Recovery (R)	

**Figure 1.** Phases of fall progress process and sub-division of the critical phase.

To develop a system for real-time detection of PONR, the process of falling in an ADL situation is conceptualized as multiple frames and each frame is mapped to the result inferred by the learning model. Then, the fall progress is expressed as a result value inferred by the learning model (e.g., the continuous output value of a fall learning model, such as 3333344444, where the value of the imbalance in our system is 3 and the falling down is 4). Through continuous result values inferred from the learning model, entering the falling down stage can be detected and the PONR time can be determined in real-time. This system was performed every 10 ms in consideration of the time required for inference of the learning model and the operating time of the system collecting data from several sensors.

2.2. COM-Based Labeling

The COM is an indicator for evaluating the walking and postural stability of the human body; it provides significantly intuitive and crucial information. Walking and posture become increasingly stable as the COM becomes closer to the BOS. This concept has been actively applied to not only analyze the posture of the human body but also to develop algorithms for controlling the posture of legged robots, especially bipedal robots, such as the zero-moment point control algorithm [18]. Thus, in this study, the labeling technique for classifying the postural stability of the human body was based on the COM, which is calculated based on the following equation:

$$x_{com} = \frac{\sum_{i=1}^n M_{x,i}}{\sum_{i=1}^n F_{x,i}} = \frac{\sum_{i=1}^n m_i \cdot x_{c,i}}{\sum_{i=1}^n m_i} \quad (1)$$

Here, $M_{x,i}$ and $F_{x,i}$ refer to the moment and force applied to the i th part of the human body, respectively. m_i is the weight of the i th part of the human body and $x_{c,i}$ is the location of the COM for the i th part of the human body. Parts of the human body refer to sections of the human body that are distinguished according to joints (e.g., the brachium and antebrachium of arms, thighs, and calves). The equation for calculating the COM in the x -direction is consistent with Equation (1) and can be applied in the same way as for calculating the COM in the y - and z -directions.

Theoretically, at least 17 IMU sensors should be installed to calculate the COM of the human body in consideration of all parts of the human body. However, as it was necessary to analyze only the postural changes according to the movement of the lower body, only five IMU sensors attached to the lower body and waist of a person were utilized to estimate the COM in this study.

The fall data were labeled based on the norm information on the COM estimated by the five IMU sensors. Figure 2 shows the cases of labeling forward and backward fall data. In this figure, the black dotted line indicates the standing phase among the activities of daily living (ADL) and is labeled as S. The yellow dotted line refers to the walking phase among the ADLs and is labeled as W. The blue dotted line refers to the imbalance phase and is labeled as I. The red dotted line refers to the falling down phase and is labeled as F. Index 1 refers to units of 10 ms. $COM(x, y)$ represents the COM based on the moving coordinates on a flat surface using the label values of the fall data. Figure 2b refers to a backward fall caused by a person standing and falling backward. The $COM(x, y)$ analysis revealed that the internal area of a circle forming the BOS boundary was divided into two sections, although the person was not walking. The $COM(x, y)$ analysis in Figure 2a revealed that the black dots in the standing phase were concentrated at one dot. These dots were not clearly distinguished because they were covered by yellow dots during the walking phase. The end of the red dotted line (i.e., the falling down phase) indicates the point where the human body collides with the ground.

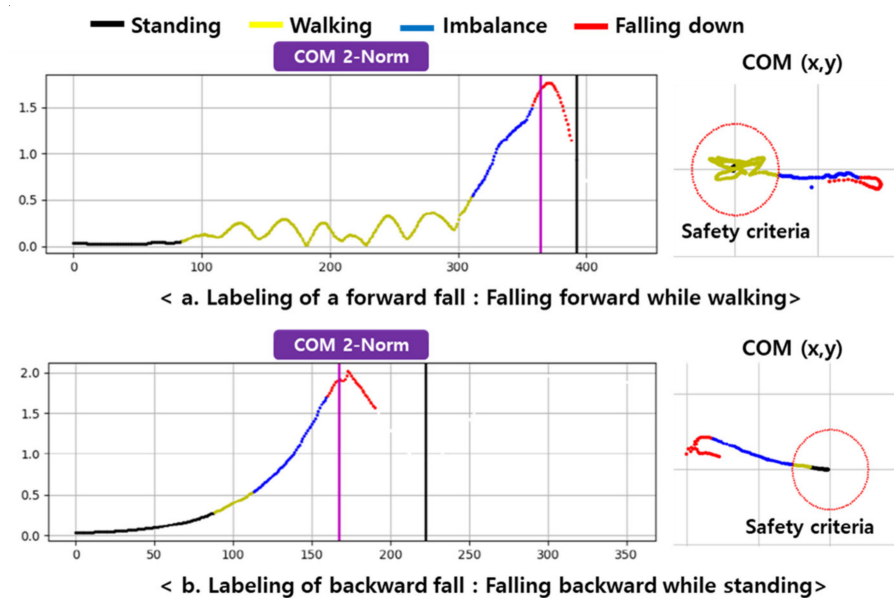


Figure 2. Examples of fall data labeling. The inverse COM is calculated and the instability increases as the value increases.

3. Data Acquisition

A mat was used to obtain vast fall data in various directions under safe conditions. A fall implementation device was also used in the experiment to obtain fall data during the walking process and verify the overall system performance. Figure 3 shows Falls data acquisition system comprising of (a) hardware and (b) software configuration of the system.

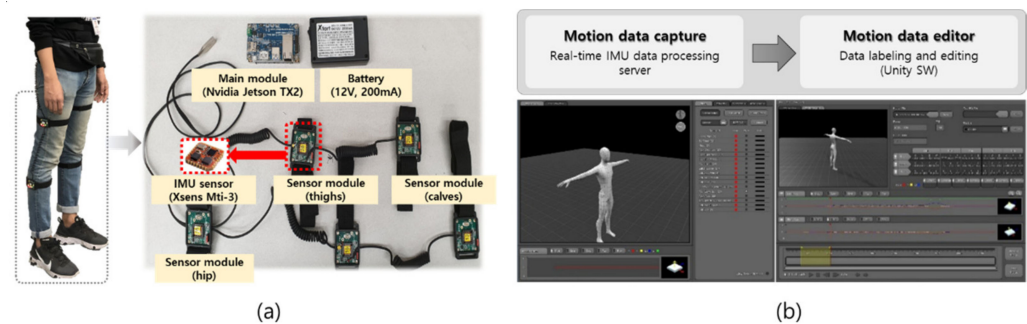


Figure 3. Falls data acquisition system; (a) hardware and (b) software configurations of the system.

3.1. Composition of the Falls Data Acquisition System

The fall data acquisition system that was developed to collect data in this study consists of a data capture device, a real-time data processing S/W program, a visual tool for previewing and editing the obtained data. Commercial data capture devices generally obtain motion data from the entire body using 17 IMU sensors. A single sensor can collect data at a rapid interval of 100 Hz. However, the capture period increased according to the volume of data to be transferred when all sensors were utilized. In addition, real-time data cannot be directly used in some cases because only data stored by a special S/W program can be used. Thus, an independent data capture device was developed in this study to obtain data in real-time. The device comprised five IMU sensors that were connected in the form of a daisy chain USB hub, enabling users to attach these sensors to their bodies more conveniently. The real-time data processing S/W program based on the server-client structure managed data collected from the five sensors in the form of packets and transmitted them through wired and wireless network communication. The visual tool for previewing and editing the obtained data prevented the collection of incorrect data by analyzing the data obtained by the data capture device, identifying motion based on skeleton-shaped images and facilitating label editing in preparation for a context that necessitates manual labeling.

The fall data acquisition system was operated based on the following processes: first, the IMU sensors of the data capture device output Quaternion, Euler, accelerometer, gyroscope, and magnetic data at 100 Hz. The real-time data processing S/W program received the aforementioned data and transferred them to the visual tool through wired and wireless networks. The visual tool monitored data in real-time and stored them as CVS files based on the preview function. Label editing was performed when needed.

3.2. Falls Data Acquisition on the Mattress

Thirty healthy female and male adults in their 20 s and 30 s were selected as experimental subjects. These participants stood on the mattress and obtained fall data from them in the forward, rear, left, and right directions and were instructed to follow some instructions. The subjects performed forward and backward falls 15 times and data were obtained from 30 subjects, 450 times in total. The 30 subjects fell sideways nine times and data were obtained from 30 subjects, 270 times in total.

Figure 4a,c show a subject falling forward and sideways, respectively. The subjects were forced to fall after they took two to three steps to obtain forward and sideways fall data. Figure 4b shows a subject participating in a real fall. The subjects were forced to fall in the same place to obtain real fall data.

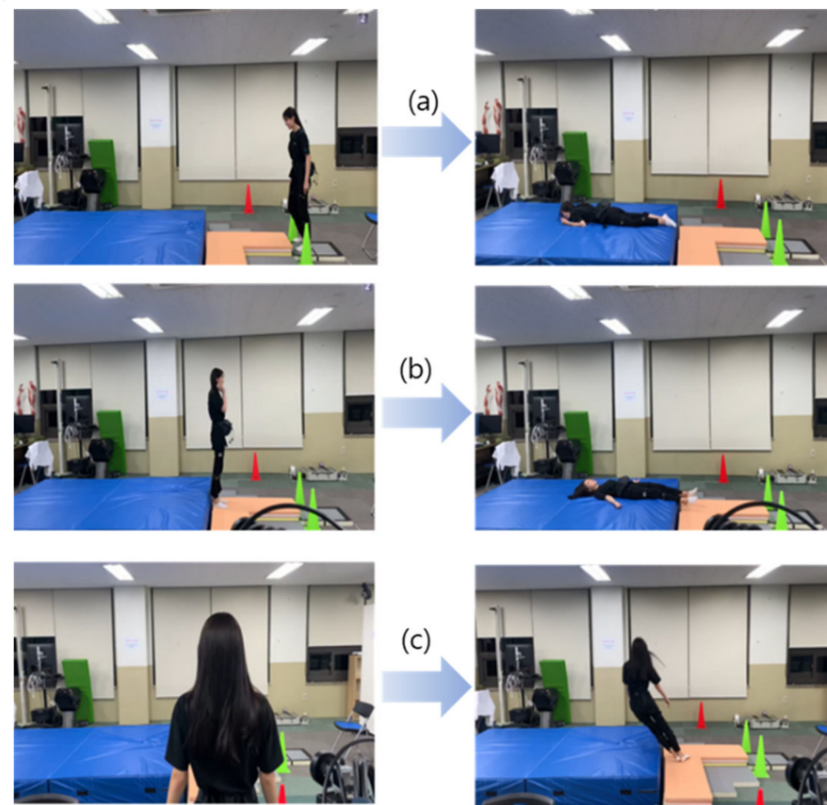


Figure 4. Falls data acquisition experiment; (a) forward, (b) backward, and (c) sideward fall experiment.

3.3. Falls Data Acquisition on the Treadmill

Fall data were acquired using a treadmill equipped with a function to induce falls. The test subjects consisted of 30 healthy adult males and females in their 20 s and 30 s. For the fall data acquisition, forward and backward falls were performed 10 times each, and a total of 300 fall data were acquired.

Figure 5 shows the treadmill developed to trigger falls. Two rails were designed to rotate independently and the rotation velocity in the forward and reverse directions was controlled to trigger falls. For forward and backward falls, control commands for changing the velocity of the developed treadmill were generated at random intervals to prevent any anticipation. To reduce the cognitive learning performance of the subjects, they were asked to listen to music using ear sets and watch videos shown in front of the treadmill. The experimental environment was then darkened. Harnesses and safety devices were used to ensure the safety of the subjects.

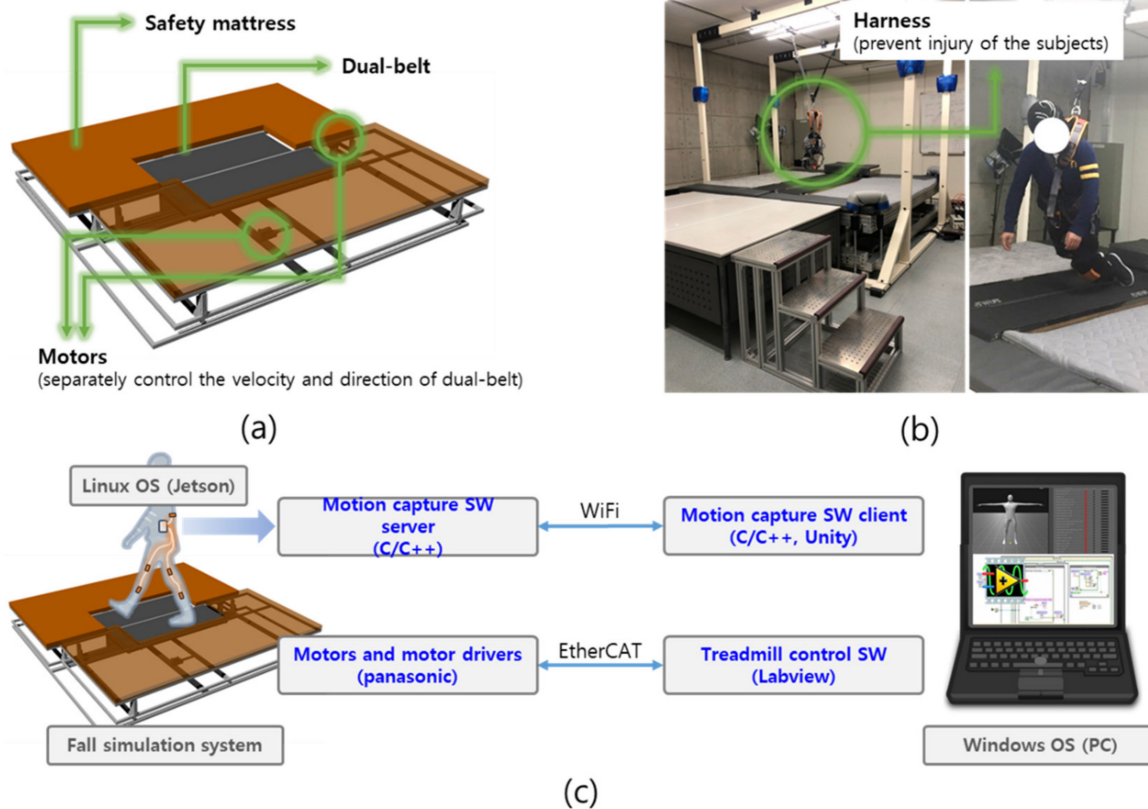


Figure 5. Dual-belt treadmill device for fall simulation experiment; (a) 3D-cad model of dual-belt treadmill device, (b) the fall simulation experimental system, and (c) data flow diagram of the system.

4. Learning and Recognition Results

For fall data learning, the LSTM algorithm as shown in Figure 6a and CNN as shown in Figure 6b were applied. In the LSTM model, the input sensor data were analyzed as much as the time step of the window size to classify the category to which it belongs. In the CNN model, the window size of the IMU sensor data and the number of sensor inputs were mapped to the height and width of the image data, as was reported by Rueda et al. in 2018 [20] and Gholamrezai et al. in 2019 [21]. This was applied to as being used as the input data that was applied to the CNN algorithm as shown in Figure 6b. Softmax was used for classification in the final output.

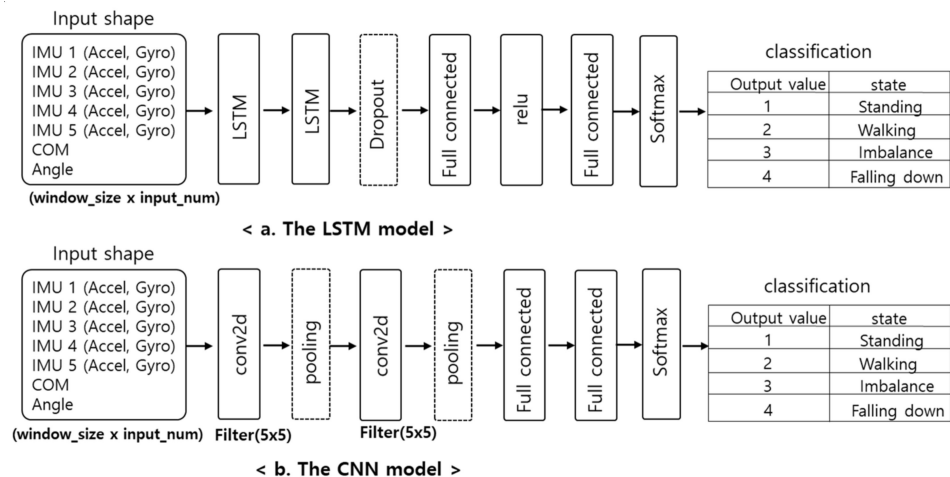


Figure 6. Learning models.

In the learning process, the window size was established based on the duration of the imbalance and falling down phases. The results of the fall data analysis used in the learning process revealed that the mean durations of the imbalance and falling phases were 420 and 250 ms, respectively. The fall data were input at intervals of 10 ms. Therefore, the time scale of the window was 250 ms. The mean duration of the falling down phase was calculated with a window size set to 25. The trend of recognition rates of the learning model was examined during the learning process, where the window size ranged from 8 to 24. The learning rate was set to 0.001 and the mean squared error was used as a loss function. ADAM (lr = 0.001, beta_1 = 0.9, beta_2 = 0.999, epsilon = 1×10^{-8}) was utilized as the optimizer.

As shown in Figure 7, tests on the learning models were conducted using data that were not applied in the learning process. A single fall case was saved as a single file. As shown in Figure 7a, the test results for each fall case were stored as CSV files. As shown in Figure 7b, the true label values and predicted values were plotted at the top and bottom, respectively, to simplify the analysis. The predicted result was correct if the colors of the bars at the top and bottom were the same (otherwise, they were incorrect).

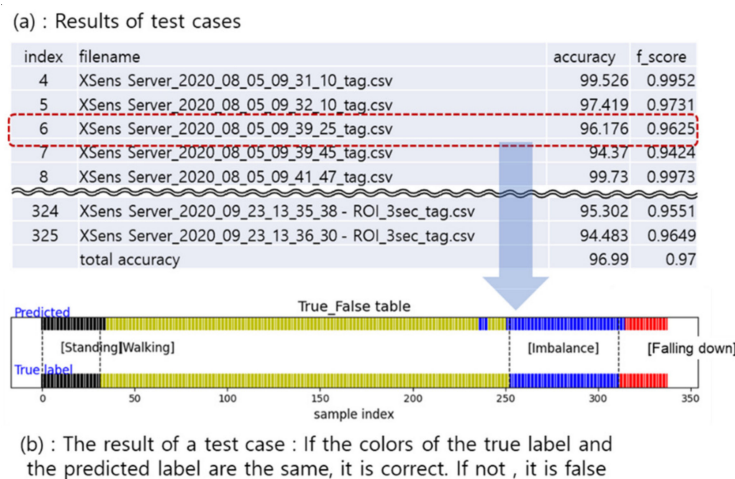


Figure 7. Recognition rates of learning models according to cases.

Regarding the input features indicated in Figure 8, the recognition rates of the learning models were measured with (blue line) and without (orange line) the yaw value of the gyro and the COM for comparison. Figure 8 shows that the recognition rate of the CNN model on the ground tended to decrease gradually owing to its greater window size. When the COM was input as a feature, its recognition rate increased. However, the recognition rate of the LSTM model is not affected by window size. When the COM was input as a feature, the recognition rate increased.

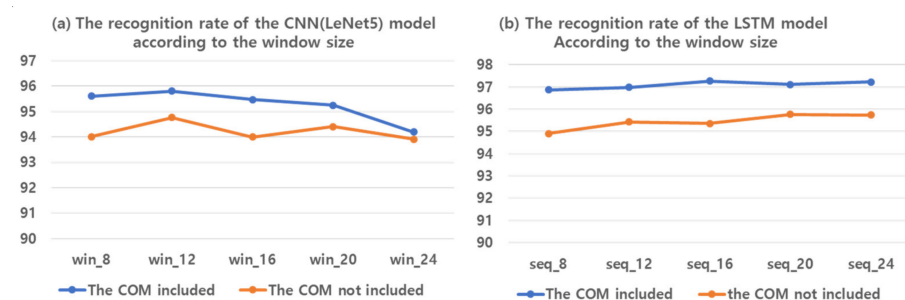


Figure 8. Results of recognition rates of CNN and LSTM models.

As shown in Figure 9, when the fall progress is viewed as a data flow, the structure of the fall data has a cascading change structure (a structure in which data in the same

state is continuous and then changed to another state section at one moment). When data containing a transition section were input to a model trained by the CNN, recognition errors mainly occurred near the transition section. In addition, the number of recognition errors increased owing to the larger window size. It was conjectured that a recognition error occurred because the information in the previous section and information in the next section coexist in the image information corresponding to the window size in the transition section. In contrast, the LSTM model generated a lower number of recognition errors in the transition section. It was deduced that the difference between the recognition rates of the CNN and LSTM models indicated in Figure 8 was caused by a difference in the degree of recognition errors in the transition section. This result is also supported by the fact that the recognition rate of the LSTM model did not decrease, despite an increase in the window size, unlike the CNN model.

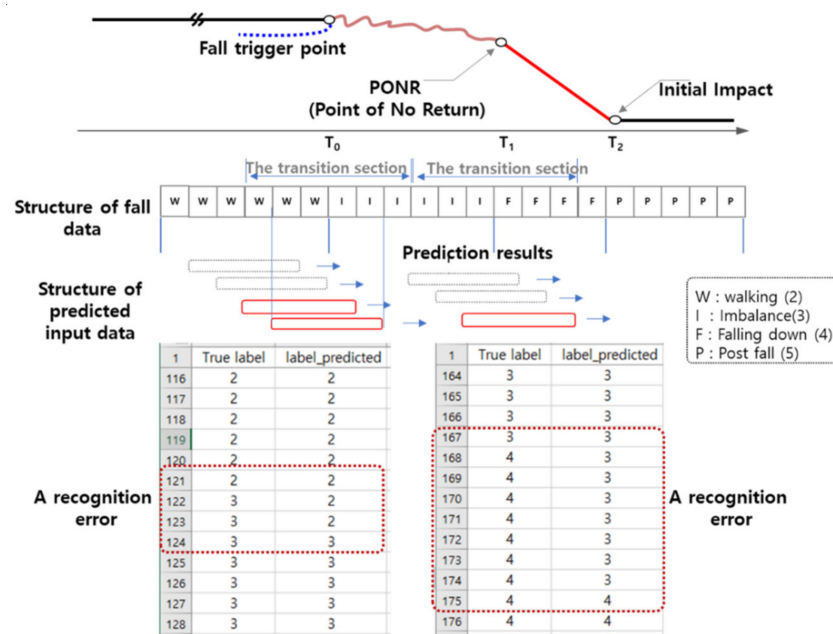


Figure 9. Recognition errors in transition sections.

5. Real-Time Implementation and Verification

The real-time execution program was operated using Python3.x, Tensorflow 2.0, and the period of data input was 100 Hz. The main platform used was Jetson TX2, and only the CPU was utilized for execution.

Figure 10 shows how to validate a fall prediction program in real-time. As shown in Figure 10a, the server collects sensor data and converts it into a transport packet. Sensor data collection may be performed in real-time from a data acquisition device, or data collected in advance and stored in CSV format may be loaded. The server transmits packet data to the client at 100 Hz. The client connects to the server, receives sensor data, and executes the fall prediction program. In the fall prediction program, the prediction result is output by inferring the input data using the pre-trained model. In Figure 10b, the true_label(1) value at the bottom means the correct answer to be predicted, the top target_pred(2), is the result inferred from the learned model by receiving the test fall data file from the server through TCP and executing the fall prediction program implemented in the multi-process structure in the client. (c) is the result of executing the model on LeNet5, a kind of CNN, on the target platform. (d) is the result of running the LSTM model on the target platform. The input data used in (c) and (d) are the same. The LSTM model generates fewer errors than the CNN model.

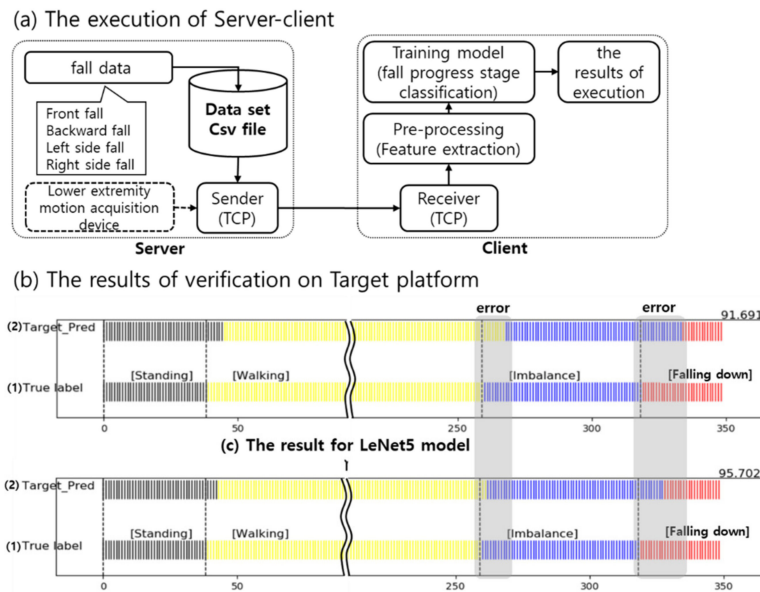


Figure 10. Model verification method and verification results.

As shown in Figure 11, the real-time execution program executed in the client has the following multiprocessor structure: (1) a processor that manages data packets received at a cycle of 100 Hz from the server, (2) a processor that converts the input format to be used in the learned model, (3) a processor that performs inference of the learned model, and (4) a processor that outputs the result of inference.

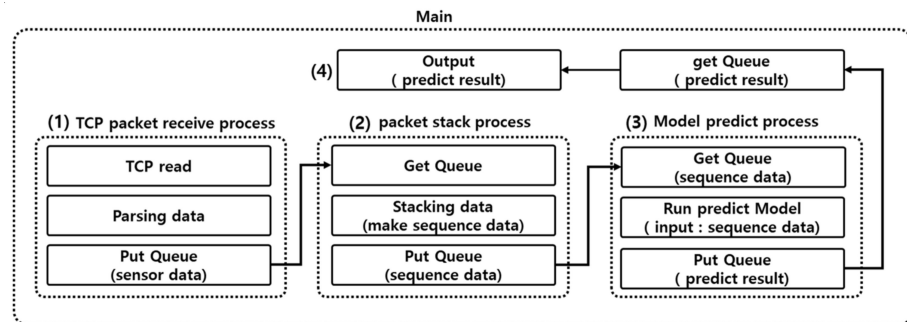


Figure 11. Real-time execution program.

Moreover, the execution times of the CNN and LSTM models that were already trained on the Jetson TX2 platform were measured, as shown in Figure 12. Based on the measurement results, it was shown that the CNN model performed faster than the LSTM model and it was confirmed that both models performed within 10 ms.

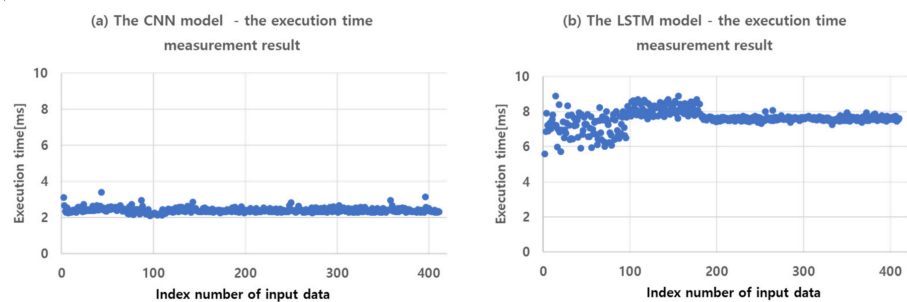


Figure 12. Real-time execution time measurement.

The fall prediction program was operated at an interval of 10 ms and determined the PONR when the imbalance stage was converted to the falling down stage; three to four prediction results were derived in a row, demonstrating that real-time fall prediction can be performed using both the CNN and LSTM learning methods. Because the CNN algorithm has a shorter execution time than the LSTM algorithm, we confirmed that it has the potential to be applied to lightweight embedded systems.

6. Conclusions and Further Research

A fall goes through the stages of fall progression, as shown in Figure 1. If the instantaneous scene can be mapped to a label value representing the fall progression stage, then the fall progression is expressed as a sequence of label values. The scene at every moment during the fall was mapped to sensor data comprising $\text{window_size} \times \text{input_num}$. In this study, considering the input data and real-time execution time, the length of each moment was set to 10 ms. The stage at which the instantaneous scene was made up of sensor data for every moment 10 ms belongs to the subdivided fall progression stage. The progress of the fall could be judged by the continuous label values determined in this way and it could also be determined whether the person was passing the PONR point. The major factors in achieving this were that more accurate labeling was possible based on the COM and that the inference execution time of the learned model was within 10 ms. To estimate the inference execution time of the learned model, the multi-process structure was utilized as much as possible and learning was performed by setting a small window size of the input data.

However, this study has several limitations. First, collecting an enormous volume of fall data that completely reflected practical situations was difficult. Furthermore, the standards used to collect such data may vary. In this regard, further research should be conducted to increase the data volume through a method [22] that can modify sensor data while preserving the labels. Another approach would be to apply a method that will enable the learning model to perform reinforcement learning of the simulated results for the human body based on sensor data obtained by the fall implementation device and generate new fall data through the learning process. Specifically, simulation based on reinforcement learning can be useful for simulating risky situations in which a fall test cannot be performed several times.

Author Contributions: Conceptualization, B.S.K., J.J., D.-W.L., Y.K.S. and H.C.S.; methodology, B.S.K., D.-W.L., Y.K.S. and J.J.; software, B.S.K., D.-W.L. and Y.K.S.; validation, B.S.K., J.J. and D.-W.L.; formal analysis, B.S.K., J.J. and D.-W.L.; investigation, Y.K.S., J.J. and D.-W.L.; resources, B.S.K., Y.K.S. and J.J.; data curation, B.S.K. and Y.K.S.; writing—original draft preparation, B.S.K.; writing—review and editing, B.S.K., D.-W.L., Y.K.S., J.J. and H.C.S.; visualization, B.S.K. and Y.K.S.; supervision, Y.K.S. and H.C.S.; project administration, Y.K.S. and H.C.S.; funding acquisition, Y.K.S. and H.C.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korean government (MSIT) (2017-0-01724, Development of A soft wearable suit using intelligent information and meta-material/structure technology for fall prediction and prevention) and by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government (2017-0-00050, Development of Human Enhancement Technology for auditory and muscle support).

Institutional Review Board Statement: The fall data used in this study were obtained with the approval of the IRB at Konyang University (project no.: KYU-2020-055-02).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Igual, R.; Medrano, C.; Plaza, I. Challenges, issues and trends in fall detection. *Biomed. Eng. Online* **2013**, *12*, 66.
2. Delahoz, Y.S.; Labrador, M.A. Survey on Fall Detection and Fall Prevention Using Wearable and External Sensors. *Sensors* **2014**, *14*, 19806–19842. [[CrossRef](#)]
3. Noury, N.; Bourke, A.K.; ÓLaighin, G.; Lundy, J.E. A proposal for the classification and evaluation of fall detectors. *IRBM* **2008**, *29*, 340–349. [[CrossRef](#)]
4. Casilari, E.; Santoyo-Ramón, J.-A.; Cano-García, J.-M. Analysis of Public Datasets for Wearable Fall Detection Systems. *Sensors* **2017**, *17*, 1513. [[CrossRef](#)]
5. Hu, X.; Qu, X. Pre-impact fall detection. *Biomed. Eng. Online* **2016**, *15*, 61. [[CrossRef](#)]
6. Chung, K. *Policy Report on a Survey on Welfare and Living Conditions of the Elderly, Paragraph 3 (Safety Conditions) Of Chapter 13 (Living Environments and Safety Conditions for the Elderly)*; The Ministry of Health and Welfare (MOHW): Sejong, Korea; Institute for Health and Social Affairs: Sejong, Korea, 2017.
7. Bourke, A.K.; O'Brien, J.V.; Lyons, G.M. Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. *Gait Posture* **2007**, *26*, 194–199. [[CrossRef](#)]
8. Tolkiehn, M.; Atallah, L.; Lo, B.; Yang, G.Z. Direction Sensitive Fall Detection Using a Triaxial Accelerometer and a Barometric Pressure Sensor. In Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Boston, MA, USA, 30 August–3 September 2011; pp. 369–372. [[CrossRef](#)]
9. Zhao, G.; Mei, Z.; Liang, D.; Ivanov, K.; Guo, Y.; Wang, Y.; Wang, L. Exploration and implementation of a pre-impact fall recognition method based on an inertial body sensor network. *Sensors* **2012**, *12*, 15338–15355. [[CrossRef](#)]
10. Choi, W.J.; Wakeling, J.M.; Robinovitch, S.N. Kinematic analysis of video-captured falls experienced by older adults in long-term care. *J. Biomech.* **2015**, *48*, 911–920. [[CrossRef](#)]
11. Ma, X.; Wang, H.; Xue, B.; Zhou, M.; Ji, B.; Li, Y. Depth-Based Human Fall Detection via Shape Features and Improved Extreme Learning Machine. *J. Biomed. Health Inform.* **2014**, *18*, 1915–1922. [[CrossRef](#)]
12. Fan, Y.; Levine, M.D.; Wen, G.; Qju, S. A deep neural network for real-time detection of falling humans in naturally occurring scenes. *Neurocomputing* **2017**, *260*, 43–58. [[CrossRef](#)]
13. Özdemir, A.T.; Barshan, B. Detecting Falls with Wearable Sensors Using Machine Learning Techniques. *Sensors* **2014**, *14*, 10691–10708. [[CrossRef](#)]
14. Xu, J.; He, Z.; Zhang, Y. CNN-LSTM Combined Network for IoT Enabled Fall Detection Applications. *J. Phys. Conf. Ser.* **2019**, *1267*, 012044. [[CrossRef](#)]
15. Serpen, G.; Khan, R.H. Real-time Detection of Human Falls in Progress: Machine Learning Approach ScienceDirect. *Proc. Comput. Sci.* **2018**, *140*, 238–247. [[CrossRef](#)]
16. Luna-Perejón, F.; Domínguez-Morales, M.J.; Civit-Balcells, A. Wearable Fall Detector Using Recurrent Neural Networks. *Sensors* **2019**, *19*, 4885. [[CrossRef](#)]
17. Sucerquia, A.; López, J.D.; Vargas-Bonilla, J.F. SisFall: A Fall and Movement Dataset. *Sensors* **2017**, *17*, 198. [[CrossRef](#)]
18. Otanasap, N.; Boonbrahm, P. Pre-impact fall detection approach using dynamic threshold based and center of gravity in multiple Kinect viewpoints. In Proceedings of the 14th International Joint Conference on Computer Science and Software Engineering (JCSSE), NakhonSiThammarat, Thailand, 12–14 July 2017; pp. 1–6. [[CrossRef](#)]
19. Tao, X.; Yun, Z. Fall prediction based on biomechanics equilibrium using Kinect. *Int. J. Distrib. Sens. Netw.* **2017**. [[CrossRef](#)]
20. Moya Rueda, F.; Grzeszick, R.; Fink, G.A.; Feldhorst, S.; Ten Hompel, M. Convolutional Neural Networks for Human Activity Recognition Using Body-Worn Sensors. *Informatics* **2018**, *5*, 26. [[CrossRef](#)]
21. Gholamrezaii, M.; Taghi Almodarresi, S.M. Human Activity Recognition Using 2D Convolutional Neural Networks. In Proceedings of the 27th Iranian Conference on Electrical Engineering (ICEE), Yazd, Iran, 30 April–2 May 2019; pp. 1682–1686. [[CrossRef](#)]
22. Kim, M.; Jeong, C.Y. Label-preserving data augmentation for mobile sensor data. *Multidimens. Syst. Signal Process.* **2021**, *32*, 115–129. [[CrossRef](#)]