

Received February 14, 2021, accepted February 24, 2021, date of publication March 10, 2021, date of current version April 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3065105

Deep Neural Network–Based Double-Check Method for Fall Detection Using IMU-L Sensor and RGB Camera Data

DEOK-WON LEE^{ID}, KOOKSUNG JUN^{ID}, KHAWAR NAHEEM^{ID},
AND MUN SANG KIM^{ID}, (Member, IEEE)

School of Integrated Technology, Gwangju Institute of Science and Technology, Gwangju 61005, South Korea

Corresponding author: Mun Sang Kim (munsang@gist.ac.kr)

This work was supported by the Ministry of Trade, Industry and Energy of Korea under Grant 10063300 and Grant 20003762.

ABSTRACT Existing methods for fall detection may not detect a fall when it occurs or may generate a false alarm when a fall does not occur. In order to overcome these limitations and detect falls with 100% accuracy, a double-check method for fall detection in elderly people via an inertial measurement unit-location (IMU-L) sensor and a red–green–blue (RGB) camera is proposed. The IMU-L sensor is a combination of an IMU sensor (accelerometer and gyroscope) and an ultrawideband signal-based location sensor; the RGB sensor is mounted on a robot. The proposed method involves detecting and confirming the fall of an elderly individual via the IMU-L sensor and an RGB image, respectively. The IMU-L sensor is worn on the body to detect falls. When a potential fall occurs, the individual’s location information is synchronized with the motion data. During detection, because of the sequential nature of IMU data, a deep learning technique called a recurrent neural network (RNN) is trained to classify falls. When the IMU indicates a suspected fall situation, the robot moves to the corresponding location and confirms whether a fall has occurred. During the confirmation stage, a convolutional neural network-based technique is applied to the RGB image data to recognize and confirm the fall. Repeated confirmed fall detections using this method classified falls more accurately than existing methods that use only an IMU sensor. We conducted a real-time experiment to validate our method using a dataset developed in a laboratory and achieved 100% accuracy in our experimental environment.

INDEX TERMS Convolutional neural network, deep learning, elderly fall, fall detection, motion data with location, recurrent neural network, transfer learning.

I. INTRODUCTION

A fall is a highly threatening situation for elderly people. A fall by an elderly person may cause serious injury or a life-threatening situation [1], [2]. Thus, it is not surprising that the World Health Organization (WHO) [2] reported that falls are the second leading cause of accidental death following traffic accidents and more than 600,000 people die each year from falls. Because the number of elderly people worldwide is expected to increase to approximately 22% (approximately 1.0 billion to 1.2 billion) by 2050, the problem of falls by elderly people is likely to become even more serious [3], [4]. Therefore, prediction in advance of a fall by an elderly person

is a very important concept. In [5]–[7], the authors presented a technique for predicting future frames of video data; this technique can help predict a fall by an elderly person. However, if a fall cannot be avoided by an elderly person, measures should be taken to ensure that the fall is detected and treated as quickly as possible. This is because elderly people cannot ask for help on their own if they are seriously injured or unconscious from a fall.

Consequently, many research teams are developing fall detection systems. The typical methods for automatically detecting falls by elderly people involve using a camera or motion sensor [8]–[15]. One common method involves detecting falls using a red–green–blue–depth (RGB-D) sensor, such as a Microsoft Kinect [16]–[29]. In fall detection methods that use a camera, the system first processes RGB

The associate editor coordinating the review of this manuscript and approving it for publication was Kathiravan Srinivasan^{ID}.

images or depth data to recognize a person and then determines whether a fall has occurred. In addition, fall detection systems that utilize joint information from humans have been developed using depth data. However, because this type of method uses a fixed RGB-D sensor, it can detect falls only within the field of view of the RGB-D sensor.

Another typical method for fall detection is to use a system with a motion sensor such as an inertial measurement unit (IMU). Using parameters such as velocity, angular velocity, direction, and acceleration recorded by the IMU, the IMU can control instruments such as robots, track human joint information, classify behavior or detect falls [30]–[47].

However, IMU-only systems make it difficult to take immediate action in response to a fall because the location where the fall occurred is unknown. Therefore, to solve this limitation [48], focused on developing a technique to measure the location of a fall. In addition to these methods [49], also predicted the fall risk using a gait analysis.

However [50]–[52], state that these existing methods can generate false alarms. For example, actions such as the IMU wearer picking up dropped objects or lying on the ground can also be misclassified as falling. In addition [52], [53], state that wearable sensors are advantageous in terms of cost, but vision sensors are better in terms of accuracy. In [54], the authors studied a fall detection method using a fused system comprising an IMU sensor, an electroencephalograph (EEG) sensor and a fixed RGB camera. If another type of sensor is fused with a vision sensor, the fall can be verified once again, and good performance can be achieved, but only falls occurring within the camera's field of view can be detected. This constraint requires a large number of cameras and PCs to be installed to achieve full coverage of a large space such as a nursing home, which, in turn, leads to increased costs.

To overcome these limitations, we use a method that simultaneously acquires a user's location data and motion data, such as acceleration and angle. Our system utilizes the location data to move a robot to the location where a fall may have occurred; the system then double-checks whether a fall occurred by using image data obtained by a camera mounted on the robot. This method reduces fall detection errors as follows: when the analysis of the motion data collected by the IMU-L sensor detects a fall, the robot moves to the fall location and acquires an RGB image using its RGB camera ten times and then analyzes the RGB image to confirm the fall situation.

This study uses a recurrent neural network (RNN) and a convolutional neural network (CNN) to process the motion data from the IMU-L sensor and the image data from the robot, respectively. RNNs are widely used for activity and gesture recognition [55]–[57], gait recognition [58]–[60], and natural language processing [61], [62], and CNNs are widely used for object recognition and detection [63]–[68]. Because motion sensor data are sequential data, they can be analyzed with the RNN, and because an RGB image is a single-frame picture, it can be analyzed using the CNN. Because CNN algorithms require long learning times and large datasets

to obtain good results, we used a transfer learning method, which can overcome the limitations of a small dataset and reduce the learning time [69]. Unlike previous studies, during the double-check step, the user's fall is confirmed using an RGB image. Because we use a single-frame image of a person lying on the floor, continuous RGB frames are not required. This is possible because the IMU sensor data are first used to determine a probable fall; then, the double-check to ensure that a fall actually occurred is performed using the image data after moving the robot-mounted image sensor to the user's location.

The novel contributions of the study described in this paper are summarized as follows.

- In existing fall detection methods that use an IMU sensor, when a fall occurs, the detection accuracy is not 100%. We propose a double-check method that can detect 100% of falls using an IMU-L sensor and an RGB camera mounted on a mobile robot. In this double-check method, when a fall is detected using an IMU-L sensor, the robot moves to the corresponding position and checks again for a fall using a mounted RGB camera. RGB images are acquired and analyzed ten times, and the accuracy is improved by deriving a label with a high probability. We use a robot that can move at a speed of 1 m/s. By achieving fall detection using RGB images within 2 seconds per image, the latency to recheck the fall is minimized.
- We compared the IMU-based fall detection performances of several different RNN architectures, including basic RNN, bidirectional RNN (Bi-RNN), long short-term memory (LSTM) [70], [71], and a gated recurrent unit (GRU) [72], [73], and applied the best RNN architecture to the fall detection process using the IMU-L data.
- We compared the performance of RGB-based fall detection using various CNN algorithms used in the ImageNet Challenge (ILSVRC) [74], including VGGNet, ResNet, DenseNet, AlexNet, SqueezeNet, and Inception v3; then, we used the best CNN architecture to build the secondary fall detection algorithm.

The remainder of this paper is organized as follows. In section II, we present related work on fall detection methods. In section III, we describe the systems and environments employed in this study. In section IV, we describe the data set utilized in this study and the algorithm selected to analyze it. In section V, we present the results of experiments conducted using the datasets and algorithms described in the previous section. Section VI provides conclusions, and section VII describes future works.

II. RELATED WORKS

In this section, we present a compilation of related work on methods for detecting falls. The section is divided into a total of 3 parts, and the approaches that use an RGB-D camera, IMU sensor, and multimodal sensors are described.

A. APPROACHES USING THE RGB-D CAMERA

The method that uses an RGB-D camera is a study on a typical fall detection method.

In [8], the authors extracted the fall pose and general pose from the depth image obtained by an RGB-D camera and classified it using a support vector machine (SVM). The accuracy of each pose classification was improved by calculating the distance between a person's center point and a floor surface. They achieved 100% recall.

The authors of [23] create a bounding box using the depth information of a person obtained from an RGB-D sensor and detect a fall using the change in the width and/or depth of the bounding box. They achieved 83.0% and 87.0% for recall and accuracy, respectively.

In [28], the authors measured the direction of the human body by connecting several joints (head, center of shoulder, spine, hip and mean point of the knee) using the human joint information measured by an RGB-D sensor. Afterward, a fall was detected by measuring the angle between the direction of the human body and the floor and the speed of change. They achieved 92.5%, 100%, and 95.8% for recall, precision, and accuracy, respectively.

B. APPROACHES USING AN IMU SENSOR

Fall detection using an IMU sensor is another representative approach.

The authors of [32] use IMU sensors (accelerometers and gyroscopes) to detect falls. They collected 4 types of falls (forward, backward, lateral and failure to sit on a chair) and 6 types of activity of daily living (ADL) data (sitting on a chair, getting up from a chair, lying down on a bed, getting up from a bed, waling and walking along a stairway). The authors detect a fall, possible fall, and ADL using sensors worn on the waist. After a certain period of time after a fall was detected, they classified fall and possible fall using the pose information of sensors worn on the waist and ankle. In this way, the authors were able to reduce the false alarm of fall detection. They achieved 95.6% recall.

In [46], the authors obtained acceleration and angular velocity using an IMU sensor worn on the waist. They collected data on nine types of falls (slipping-backward, walking-tripping-forward, jogging-tripping-forward, sittingdown-backward, sitting-backward, forward, backward, lateral and twist) and 14 types of non-falls (walking, jogging, squatting, waist bending, stumbling while walking, jogging in place, jumping, ascending and descending stairs, slowly sitting up on a stool, quickly sitting up in chair, trying to get up and collapsing into a chair, lying, slowly sitting up on a low-height mattress and quickly sitting up in a low-height mattress) and designed a classifier using an artificial neural network (ANN). The authors achieved 99.86% classification accuracy for falls and non-falls.

Representative fall detection technologies that use IMU sensors commercialized in recent years include “Apple Watch 4(or later version), Apple, Inc. [75]”, and “Galaxy

Watch 3, Samsung, Inc. [76]”. These devices detect a fall and generate an alarm if the user of the device does not move for a certain period of time after a strong shock is delivered to the user while wearing the watch.

There are other studies on fall detection using smartwatches. In [47], the authors proposed an application that detects falls using accelerometer data collected from commercially available smartwatches. They utilized Microsoft Band 2 to collect 7 types of activities of daily living data (sitting, getting up, jogging, throwing an object, waving, taking a drink, and going upstairs and downstairs) and 4 types of fall data (forward, backward, left-side, and right-side). They classified data classes using naive Bayes, SVM, and GRU models and obtained 100% recall and 85% accuracy when using the GRU.

C. APPROACHES USING THE MULTIMODAL SENSORS

Research has also been conducted using multimodal sensors to reduce false alarms and increase measurement accuracy. The multimodal sensor method has the advantage of improving the accuracy of fall detection by redundant inspection of falls. Alternatively, the detection area that cannot be detected using only one sensor is supplemented with different sensors.

In [11], the authors conducted a study to detect falls using an accelerometer and an RGB-D sensor. In this study, the accelerometer data were by the fuzzy engine to infer motion, and the depth data of the RGB-D sensor was used by the fuzzy engine to infer posture. In addition, inference is not performed on every frame but is executed when the results of the threshold analysis of the accelerometer data predict a possible fall. They achieved 100%, 93.75%, and 97.14% for recall, precision, and accuracy, respectively.

In [54], the authors collected and published data on six types of ADL (walking, standing, picking up something, sitting, jumping and laying) and five types of falls (forward using hands, forward using knees, backward, sitting in a chair and sideward). A total of 17 healthy subjects participated in this experiment. They simultaneously collected data using an accelerometer, gyroscope, light value sensor, electroencephalography (EEG) headset, and RGB cameras, and validated their performance using machine learning models. The authors combined and verified the data obtained from each sensor in seven combinations of modalities, and the best results were obtained when the video frames acquired by the RGB camera were input and verified by a classifier using the CNN algorithm (71.3% recall, 71.8% precision and 95.1% accuracy).

III. MATERIALS

In this section, the IMU-L sensor, robot, and RGB camera used to detect falls are described. In addition, we also describe the environment in which the system is installed and the characteristics of the software to run them.

A. IMU-L SENSOR SETUP AND MOTION DATA ACQUISITION

The IMU-L sensor used in our system uses an ultrawideband (UWB) signal to determine the location of the sensor worn by the user. This IMU-L sensor was developed at the Center for Healthcare Robotics of the Gwangju Institute of Science and Technology (GIST, Republic of Korea). In this sensor, the DWM1000 module from Decawave Corp. was used to transmit/receive UWB signals, and the MPU9250 from TDK Invensense Corp. was used as the IMU sensor.

Four anchors and one master anchor must be installed for location measurement. The master anchor is responsible for controlling the blink rate of the two IMU-L sensors and sending the collected ranges of anchors 1–4 to a personal computer (CPU: Intel Core i5-7260U; RAM: 8 GB; SSD: 120 GB) via a serial port. Then, the PC computes the location of the sensor based on the multilateration method using the strength of each sensor’s signal that reaches each anchor. The IMU-L sensor data were acquired at a rate of 16 Hz, and the sensor signal has a stable reach of up to 30 m. In this study, the size of the space consisting of 4 anchors installed to collect IMU-L sensor data is 9.5 m × 6 m, and 5 anchors (including a master anchor) are installed at a height of 2.6 m from the ground. The error range of the positional accuracy of the IMU-L sensor within this space is ±20 cm. IMU-L sensors are mounted on both the user and the robot. Fig. 1 shows the setup of the IMU-L sensor system.

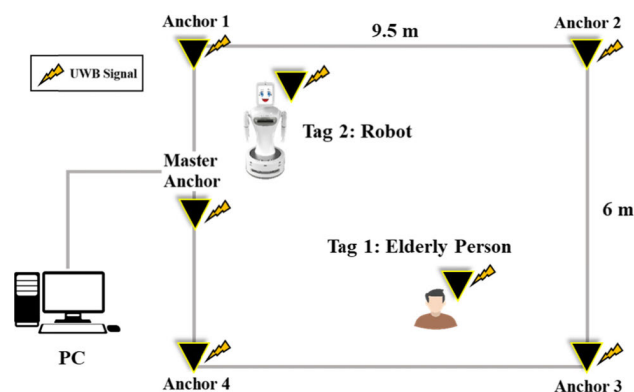


FIGURE 1. Setup of IMU-L sensor system.

The users wear the IMU-L sensor on their shoulder. Although other studies have explored activity recognition with IMU sensors worn at various positions, such as on the arms, torso, and legs, the optimal position for IMU sensors has not yet been defined; however, we found the shoulder to be a good sensor position in our experiments because less physical interference occurred when a user fell than when the sensor was worn at any other position. Fig. 2 shows the IMU-L sensor and a user wearing it.

Previous research on fall detection using only motion data, such as acceleration or angular velocity data from an IMU sensor, has shown that actions such as picking up an object

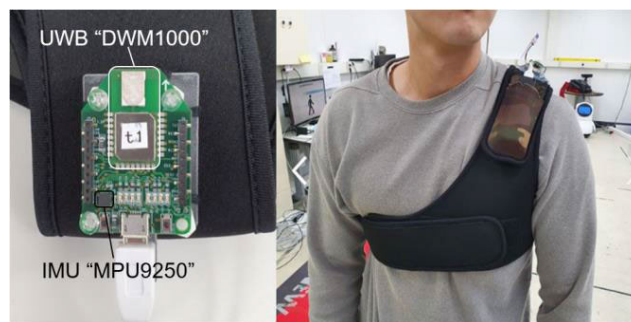


FIGURE 2. IMU-L sensor (left), and user wearing the sensor (right).

from the floor can be misclassified as a fall [51]. Similarly, actions such as lying on the ground or jumping can also be detected as falls. When the range of categories detected by an IMU-L sensor is small, a high possibility exists that other common actions will be erroneously detected as falls. Therefore, to overcome these limitations, we propose applying the IMU-L data to classify the various activities that may occur in daily life based on previous research [40], [77].

We collected IMU-L data on eleven types of actions, including four types of falls, two types of lying down, jumping, standing, walking, sitting, and picking up an object from the ground, as shown in Table 1.

TABLE 1. IMU Datasets Used to Evaluate IMU-Based Fall Detection.

Category	Subclass	Per Each Subject	Total (5 Subjects)
Falling	Falling forward	150	750
	Falling on knee	150	750
	Falling on hip	150	750
	Falling backward	150	750
No Falling	Lying supine	150	750
	Lying prone	150	750
	Picking up an object	150	750
	Standing	150	750
	Walking	150	750
	Jumping	150	750
	Sitting	150	750
Total		1,650	8,250

We collected IMU-L data on the eleven activities at 100 frames per cycle because at least 100 frames of data are required during one activity cycle. IMU-L data were obtained via laboratory experiments, and five males with no abnormalities in health participated in the experiment (age: 32 ± 7 , height: $172.5 \pm 5.5cm$, and weight: $80 \pm 19kg$).

Five people participated in the data collection, and each person performed each of the eleven activities 150 times. Therefore, we collected a total of 8,250 datasets (five subjects × eleven activities × 150 samples). Examples of

acquisition experiments and samples of acquired IMU-L data are shown in Appendix B.

B. MOBILE ROBOT SYSTEM TO DOUBLE-CHECK FALL DETECTION

In this study, when a potential user fall is detected by IMU-L sensor data analysis, a mobile robot system is used to double-check the fall situation of the user using an RGB camera. The robot used was a Silbot-3 [78]–[80], developed and sold by Robocare Corp. of South Korea.

The Silbot-3 is fully compatible with the Robot Operating System (ROS) [81]–[83]. Because Silbot-3 uses omnidirectional wheels, it can move in any direction at a maximum speed of 1 m/s and is equipped with an ASUS Xtion Pro Live [84], [85] RGB-D sensor that is positioned 76 cm from the ground. The ASUS Xtion Pro Live sensor includes an RGB camera whose maximum image resolution is 640×480 pixels. We utilized the OpenNI-based `image_view` package in the ROS to acquire RGB images at a rate of 30 Hz.

C. DATA ACQUISITION USING THE ROBOT-MOUNTED RGB-D SENSOR

Using the robot, we acquired two categories of RGB images: “Fall” and “Non-Fall” events. In “Fall” datasets, we increased a variety of situations by including a fallen person with people who were standing or sitting on a chair and several objects. Image samples for each category are shown in Fig. 3.

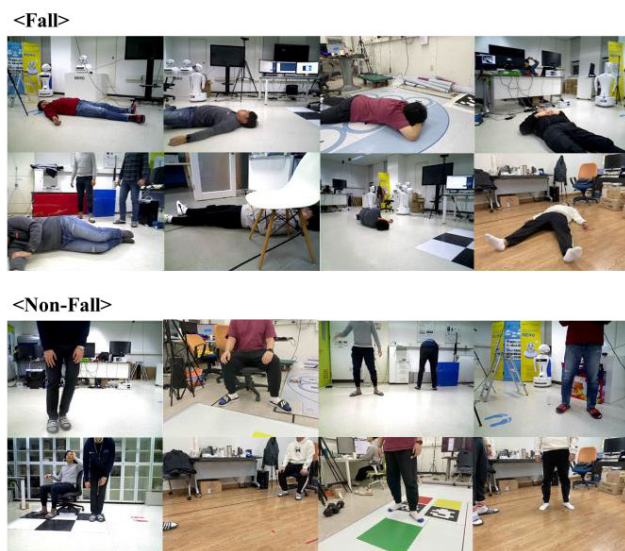


FIGURE 3. Image samples for each category.

The fall category includes images where the user is lying on the ground. This category includes both full and partial body images and contains images in various poses. In addition, to ensure the diversity of the data, the participants moved to different locations with floors of different colors and patterns, and we gathered data at a distance of 1 m to 2.5 m around the participants. The non-fall category contains images of

situations where the user is not lying on the ground, such as when one or more users are standing or sitting on a chair. The RGB dataset contains images of five subjects totaling 5,000 images (five subjects \times two categories \times 500 samples). The subjects who participate in RGB image data collection are the same subjects for IMU-L data collection. The RGB image dataset is summarized in Table 2.

TABLE 2. RGB Image Dataset for Learning using CNN Algorithm.

	Per Each Subject	Total (5 Subjects)
Fall	500	2,500
Non-Fall	500	2,500
Total	1,000	5,000

D. SYSTEM INTEGRATION USING ROS

ROS is a system for controlling several components of a robot from a system, such as a PC. ROS consists of several independent nodes, all of which can communicate with each other through publish/subscribe messages. ROS is open source, does not require multiple PCs to operate multiple nodes, and has the advantage of being able to run each node with a different OS. There are a total of 4 independent nodes in this system, and they exchange messages with each other by registering with the ROS Master. First, the ‘IMU-L Sensor Node’ publishes the message received from the IMU-L sensor connected to the serial port as an ROS message. The ‘Device Node’ controls the wheel of the robot, and the ‘Camera Node’ publishes the RGB image received from the camera connected to the robot as an ROS message. Finally, ‘Fall Detection Double-check Node’ detects a fall using data subscribed from ‘IMU-L Sensor Node’ and ‘Camera Node’. The ROS integration architecture developed in this study is shown in Fig. 4.

IV. METHODS

In this section, we first explain our proposed fall detection method using the IMU-L sensor and the robot’s RGB sensor. We also describe how the IMU-L and RGB data are analyzed using the RNN and CNN algorithms, respectively.

A. PROPOSED METHODS TO DOUBLE-CHECK FALLS

We propose a method to double-check falls initially detected using an IMU-L sensor worn by the user with an RGB sensor mounted on a robot. The process designed to detect falls is shown in Fig. 5; the details are presented as follows:

- A subject wearing the IMU-L sensor randomly repeats the eleven actions mentioned in Table 2: four types of falling, two types of lying down, jumping, standing, picking up an object, sitting, and walking. Each of the four subjects repeats every action a total of 150 times for every 100 frames of the IMU-L sensor. The collected datasets for every 100 frames of the IMU-L sensor are input to the trained RNN-based fall detection model, and

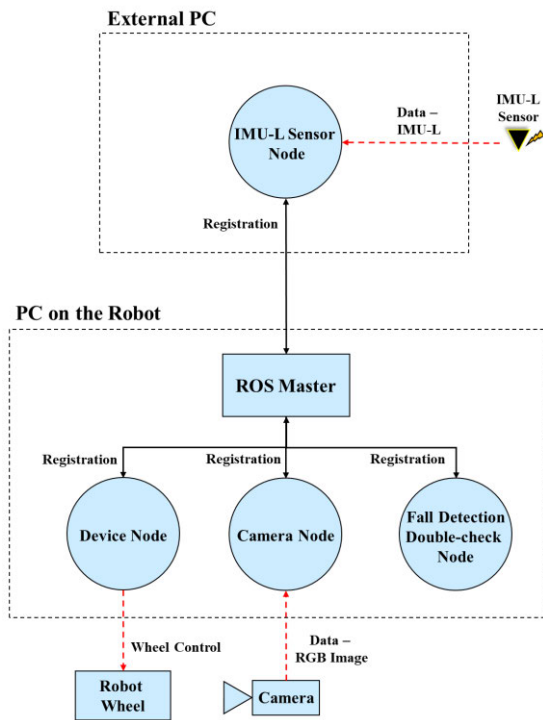


FIGURE 4. ROS integration architecture.

the results are monitored. If the user's fall is not detected, the IMU-L sensor data are used to continuously track the user's movements in real time.

- If a fall is detected in the first step, the robot moves to the corresponding area using the location information from the IMU-L sensor.
- After moving to the user's expected fall area, the robot acquires images using the RGB sensor mounted on the robot.
- The RGB images are input into the CNN algorithm and used to determine whether a fall truly occurred (double-check).

When a fall is detected in the double-check step, our system takes action such as sending a rescue signal.

B. RNN NETWORK FOR CLASSIFICATION OF SEQUENTIAL MOTION DATA

The human motion data from the IMU sensor used in this research are sequential, time-dependent data. The IMU sensor data collect three-axis acceleration data and three-axis gyroscopic data. Because human activities represented by the IMU sensor data are not uniform in length or in time of occurrence, we explored using the RNN family of algorithms to classify human activities. In addition, the fall detection performance of linear regression, logistic regression, and multilayer perceptron (MLP) algorithms was compared with that of the RNN family of algorithms.

Because the length of human activities varies, it is difficult to predict the exact duration of any given activity.

Furthermore, the basic RNN algorithm does not fully cover long-term dependencies [86]. This means that when learning from data using a basic RNN, if the distance between the relevant information and the point used is long, the gradient gradually decreases during backpropagation, which reduces the learning ability. Therefore, we also tested improved RNN architectures such as the LSTM and GRU models to overcome this disadvantage.

An LSTM is a special type of RNN that enables learning that involves a long dependency period. An LSTM is a structure that adds a cell state to an RNN's hidden state. The cell state is updated using input and forget gate values. Because the cell state acts as a type of conveyor belt, the gradients propagate relatively well for quite some time even after the state has elapsed. A GRU performs similarly to LSTM and is designed to overcome the long-term dependency weakness similarly to the method used in an LSTM. GRU uses reset and update gate values to control the amount of information, but its structure is different from that of an LSTM. A GRU performs almost identically to LSTM but has the advantage of a much simpler structure. In this study, we compared the performances of basic RNN, Bi-RNN, LSTM, and GRU models, all of which are representative algorithms of the RNN family, to analyze user motion data. Additionally, we analyzed motion data via linear regression, logistic regression, and MLP.

The structure of the IMU-based fall detection algorithm is shown in Fig. 6. The acceleration and gyroscope values (a total of six values) form the input data to a four-layer RNN structure. The hidden state values then pass through fully connected and softmax layers for classification. In the fall detection task, correctly recognizing a fall is much more important than recognizing other activities. If a fall were misclassified as a non-fall, a serious situation could result, and the proposed double-check method would be rendered useless. Therefore, we trained the model by maximizing the fall detection accuracy, although this might cause other activities to be misclassified as falls more frequently. We gave more weight to the fall datasets during the training as follows:

$$\text{softmax}(p_i) = \frac{e^{p_i}}{\sum_j e^{p_j}} \quad (1)$$

$$l = - \sum_{i=1}^2 w_i y_i \log(\text{softmax}(p_i)), \quad (2)$$

$$= -w_1 y_1 \log(\text{softmax}(p_1)) - (1 - y_1) \log(1 - \text{softmax}(p_1)), \quad (3)$$

where p_i , l , w_i and y_i denote the output of the fully connected layer, the loss for a single dataset, and the weight assigned to the dataset, respectively. Softmax is the most commonly employed component for efficiently training a deep neural networks-based classifier. Because the proposed method is a binary classification method, the weighted softmax cross-entropy loss in (2) can be transformed into (3). The p_1 and p_2 terms represent the confidence associated

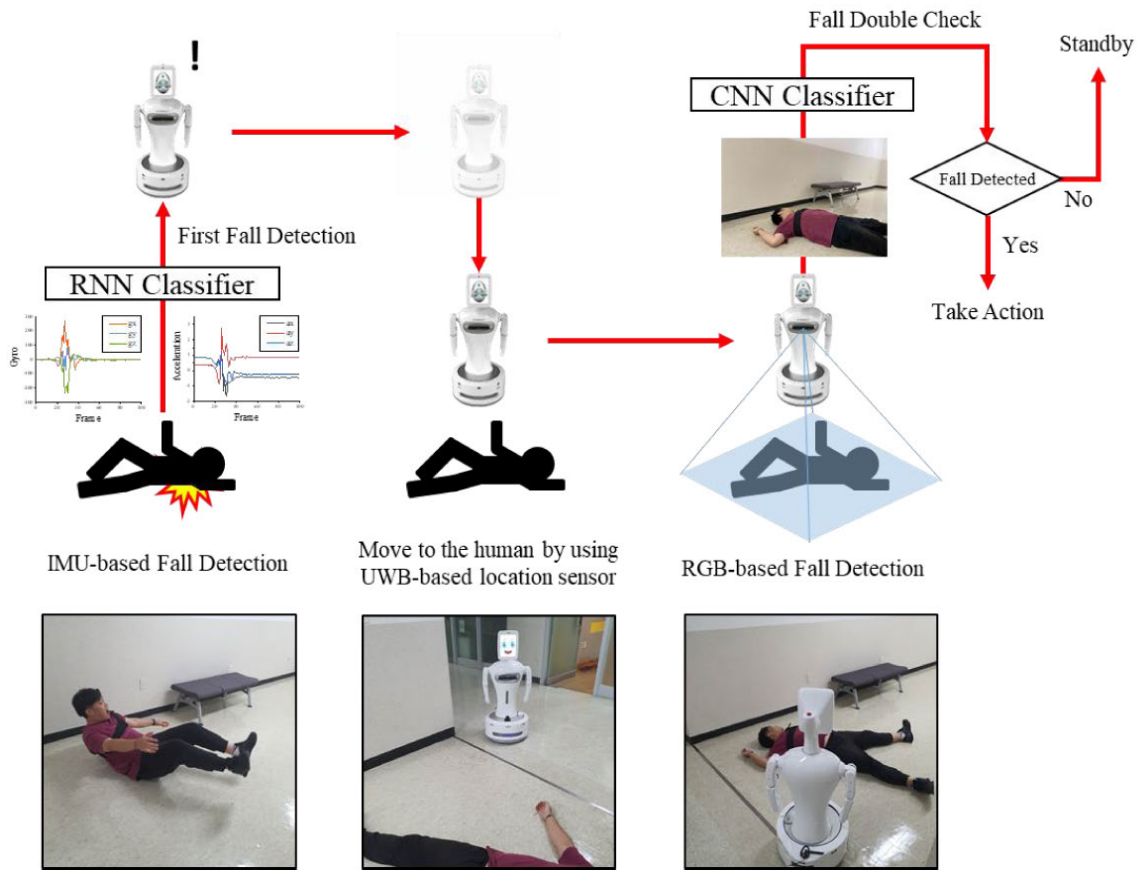


FIGURE 5. Scenario of fall detection with proposed double-check method.

with fall and non-fall activities, respectively. To give more weight to the fall datasets, w_1 (weight for the fall category) should be larger than w_2 (weight for the non-fall category). To determine the optimal weight, we evaluated the learning performance using leave-one-person-out cross-validation (LOPO CV). LOPO CV is a kind of k-fold cross-validation. In this study, we applied data from four subjects for training and one subject for testing. We performed a total of 5 cross-validations and calculated the accuracy by taking the average of the sum of each accuracy. As seen in Table 3, the best performance value is when an LSTM with a weight of 80:1 (fall:non-fall) is used. Therefore, we set the value of w in (3) to 80:1 for the LSTM.

C. CNN FOR CLASSIFYING RGB IMAGE DATA

We conducted a double-check to confirm a user’s fall when a fall was detected from the user’s IMU-L sensor data. The data used in the double-check step are RGB image data, and the process analyzes the form of a person shown in a single RGB image. A CNN algorithm is suitable for analyzing such single-frame images.

Due to fall-focused training, non-fall events might be more frequently classified as falls; however, we solve this problem by using secondary fall detection based on RGB images.

In the CNN algorithm, the convolutional and pooling layers serve to extract features from the input data. A convolutional layer is an essential element that reflects the activation function after filters are applied to the input data.

Including pooling layers after the convolutional layers is optional. The last part of the CNN algorithm adds a fully connected layer for image classification. Between the layers that extract the image characteristics and the layers that classify the image, a flattened layer is placed to form an array of data in the form of the image. A CNN is a high-performance deep learning algorithm, but it requires a substantial amount of training data and lengthy training times. In fact, training a large CNN typically requires more than a few days, even on high-performance computers. Therefore, we performed fine-tuning, a transfer learning method, using a pretrained model to overcome these disadvantages. Fine-tuning refers to a method of transforming the architecture for a new purpose (according to task-specific image data) based on a model that has been pretrained and updating the training from the already trained model weights, as shown in Fig. 7. Using this approach, we did not use the acquired RGB images to train a new model de novo; instead, we used the 1K-class Imagenet dataset [87], which is published in Pytorch, and fine-tuned a pretrained CNN model. Class labels of the 1K-class Imagenet dataset are shown in [88].

TABLE 3. Fall Detection Accuracy Performance Using IMU Data According to Learning Weight (%).

Weight (fall: non-fall)	Test Subject	Linear Regression	Logistic Regression	MLP	Basic RNN	Bi-RNN	GRU	LSTM
10 : 1	Average	48.58	48.90	49.66	39.34	41.44	36.38	39.25
20 : 1	Average	44.58	42.02	46.97	34.81	34.97	42.95	45.96
30 : 1	Average	42.59	39.27	45.34	40.00	37.53	46.46	51.23
40 : 1	Average	41.51	39.11	44.80	37.65	34.54	51.34	48.53
50 : 1	Average	40.47	38.98	44.29	34.97	35.84	51.31	51.59
60 : 1	Average	40.30	39.33	44.14	34.35	34.87	49.49	56.20
70 : 1	Average	39.92	39.42	43.33	38.45	37.07	47.29	54.02
80 : 1	Average	39.80	39.50	43.86	38.11	37.72	52.85	58.55
90 : 1	Average	39.63	39.54	43.05	36.39	38.51	47.37	57.51
100 : 1	Average	39.51	39.53	42.87	37.83	36.95	48.59	58.36

Fig. 8 compares the results of training a new CNN model de novo with the results of fine-tuning transfer learning. As Fig. 8 shows, the system converges to better learning results with fine-tuning than without fine-tuning.

We performed fine-tuning using various CNN models whose performance abilities have already been demonstrated in the ImageNet Challenge. The first algorithm used was AlexNet, which is considered the first CNN model to produce meaningful results. The dropout technique used in this model has become standard practice in this field. Since AlexNet was first proposed, attempts have been made to deepen it by adding layers or to augment each individual layer via algorithms such as VGGNet and Inception. However, deeper models may experience the vanishing gradient problem, in which the gradient is not transmitted well during backpropagation. To solve this problem, the ResNet and DenseNet models included a new type of block, called a residual block, that allows the gradient to pass through. In this study, we attempted to improve the performance of the fall detection method using fine-tuning, which allowed us to reuse the pretrained CNN models described above.

V. EXPERIMENTAL RESULTS

In this section, we present the experimental results obtained via the proposed methods. First, we evaluated the performance of the IMU-based fall detection method using an RNN-based classifier. Then, we assessed the performance of RGB-based fall detection using various CNN algorithms and compared the results. Finally, we evaluated the proposed double-check method, which uses the best IMU-based fall detection RNN model and the best RGB-based fall detection CNN model. In all the experiments, we calculated the model accuracy, precision, sensitivity, and F1-score as [89]–[91]:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (4)$$

$$Precision = \frac{TP}{TP + FP}, \quad (5)$$

$$Recall = \frac{TP}{TP + FN}, \quad (6)$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \quad (7)$$

where TP, TN, FP, and FN denote true positives (classifying a fall as a fall), true negatives (classifying a non-fall as a non-fall), false positives (classifying a non-fall as a fall), and false negatives (classifying a fall as a non-fall), respectively.

A. IMU-BASED FALL DETECTION

The proposed double-check method employs IMU-based fall detection as the first step. We developed four RNN-based binary classifiers (basic RNN, bi-RNN, LSTM, and GRU) and three non-RNN-based binary classifiers (linear regression, logistic regression and MLP). We input the 8,250 IMU data samples (five people \times eleven activities \times 150 datasets) into the binary classifiers. We fixed the number of data frames at 100. We verified the performance by LOPO CV. The IMU-based fall detection results are shown in Table 3.

Among the seven binary classifiers, the LSTM-based classifier achieved the best accuracy. The GRU-based classifier yielded results similar to those of the LSTM, but its performance was slightly poorer. The linear regression, logistic regression, MLP, basic RNN and bi-RNN models exhibited poor fall detection performance. The GRU and LSTM consider more variables than do the linear regression, logistic regression, MLP, basic RNN and bi-RNN, and their structures are more complicated. Therefore, it is easier for them to classify long-term data, and they achieve better performance in IMU-based fall detection.

In IMU-based fall detection, sensitivity is the most important metric because it is much more important to classify a fall as a fall than to classify a non-fall as a non-fall. Therefore, we deliberately assigned a higher weight to falls during the training. Note that it was expected that the total accuracy would be decreased as a result. Table 4 shows the accuracy,

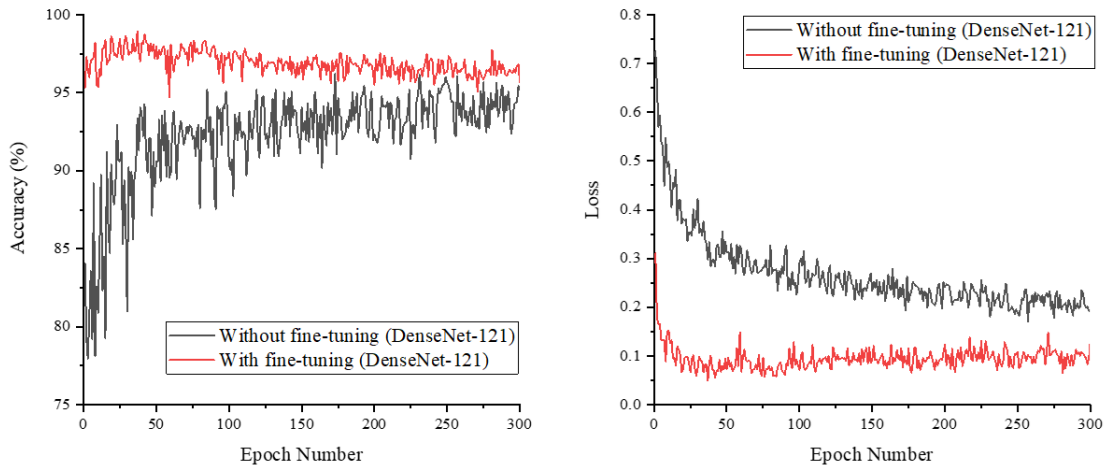


FIGURE 8. With fine-tuning results versus without fine-tuning results - accuracy (left) and loss (right).

TABLE 4. Performance of Fall Detection Using IMU Data (%).

	Linear Regression	Logistic Regression	MLP	Basic RNN	Bi-RNN	GRU	LSTM
Accuracy	48.58	48.90	49.66	40.00	41.44	52.85	58.85
Precision	38.76	38.07	39.55	36.11	36.62	41.94	45.17
Recall	88.36	80.99	91.06	99.83	99.57	100.00	100.00
F1-Score	53.88	51.79	55.15	53.04	53.55	59.10	62.23

precision, recall, and F1-score of the experiments that showed the highest accuracy in seven binary classifiers.

We were able to achieve a sensitivity of 100% using the GRU-based and LSTM-based classifiers, of which LSTM-based classifiers achieved higher accuracy. On the other hand, the precision was only 45.17%, which meant that there was a 54.83% probability that the model would classify the input IMU data as indicating a fall when no fall had actually occurred. Consequently, a fall detection system developed using only an IMU-based method will waste time and energy because of false alarms. To overcome these problems, we use IMU-based fall detection only for the first check. When a fall is detected using the IMU-based fall detection model, the system does not activate an alarm. The final decision rests with the secondary fall detection model, which uses the RGB data and the CNN-based classifier.

B. RGB-BASED FALL DETECTION

After performing IMU-based fall detection, we use an RGB image to conduct the second step in our double-check method. We developed a classifier that differentiates between fall and non-fall images by testing six CNN-based models: Inception-v3, AlexNet, SqueezeNet, DenseNet, VGGNet, and ResNet. Specifically, we used the DenseNet-121,

DenseNet-161, and DenseNet-203 models; the VGG-11, VGG-13, VGG-16, and VGG-19 models; and the ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152 models. We used a total of 5,000 (five subjects × two states × 500 datasets) RGB image samples to train and validate the CNN models. CNN-based fall detection was also subjected to LOPO CV, in which 4,000 data points were applied for training and 1,000 data points (from the subject not included in the training set) applied for the test. The entire process was repeated five times (for a total of 5 subjects). The fall detection results based on the RGB images are shown in Table 5.

Regardless of which model that we employed, we almost achieved minimum performance accuracies of 90%, but DenseNet-121 achieved an accuracy of 97.72%, which was the best result. DenseNet performed better than the Inception, AlexNet and SqueezeNet models because it uses residual blocks to reduce the residuals. Specifically, DenseNet-121 exhibited excellent performance; it had fewer vanishing gradient problems than the other DenseNet models because it has shallower layers than those models.

C. DOUBLE-CHECK FALL DETECTION

To verify the performance of the double-check method, we conducted experiments using a new dataset in which the

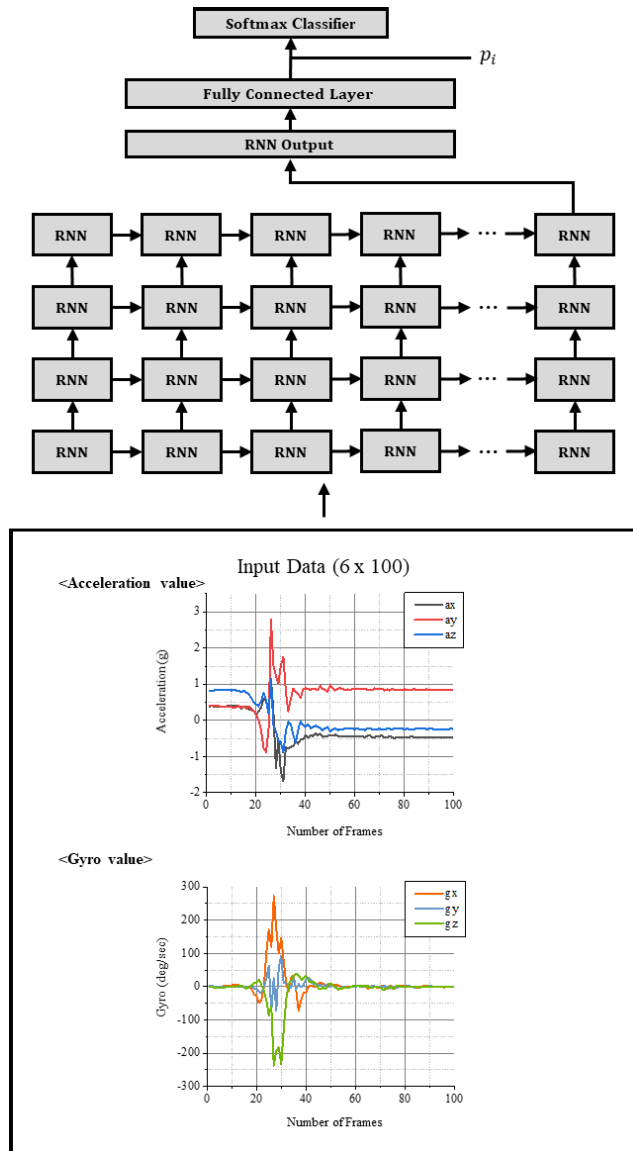


FIGURE 6. Structure of IMU-based fall classification.

samples had not been used previously. First, 1,650 IMU sensor samples (five subjects × eleven activities × 30 datasets) were used to detect fall and non-fall events. In each case, the RGB data were obtained along with the IMU sensor data. Fall detection using RGB images is the final step in detecting falls in the proposed method. Therefore, fall detection using RGB images was performed ten times to minimize errors at this stage. This was possible because fall detection using RGB images is performed using a single RGB image.

When using the IMU data to detect a fall, we adopted a higher threshold for detecting a fall; therefore, cases occurred in which a non-fall situation was detected as a fall. As shown in Table 6, when only the IMU data were employed, 662 of the 1,050 non-fall situations were detected as fall situations (FP).

We solved this problem by applying the proposed method. Under the double-check fall detection method, when a fall

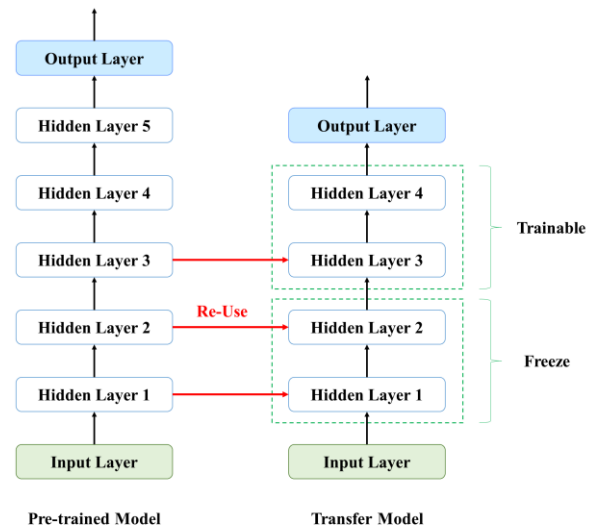


FIGURE 7. Transfer learning method.

TABLE 5. Performance of Fall Detection for Various CNN Models (%).

CNN Model	Accuracy of Each Test Subject's Average
Inception	95.92
AlexNet	58.65
SqueezeNet	80.06
DenseNet-121	97.72
DenseNet -161	96.62
DenseNet -201	96.07
VGG-11	96.96
VGG-13	96.08
VGG-16	94.59
VGG-19	92.36
ResNet-18	94.18
ResNet-34	88.93
ResNet-50	89.77
ResNet-101	85.63
ResNet-152	82.91

TABLE 6. Confusion Matrices for IMU-Based Fall Detection.

Fall Detection using only IMU data	Actual Fall	Actual Non-Fall
Predicted Fall	600 (TP)	662 (FP)
Predicted Non-Fall	0 (FN)	388 (TN)

was detected from the IMU sensor data, an RGB image was used to double-check whether a fall had occurred. Since RGB-based fall detection is performed only when a fall is detected in IMU-based fall detection, a total of 1,262 RGB-based fall detections are performed, and the results are shown

TABLE 7. Confusion Matrices for RGB-Based Fall Detection (Results after Performed Once and 10 Times).

Fall detection using RGB data (performed once)	Actual		Fall detection using RGB data (performed 10 times)	Actual	
	Fall	Non-Fall		Fall	Non-Fall
Predicted Fall	569 (TP)	11 (FP)	Predicted Fall	600 (TP)	0 (FP)
Predicted Non-Fall	31 (FN)	651 (TN)	Predicted Non-Fall	0 (FN)	662 (TN)

in Table 7. However, as shown on the left side of Table 7, when fall detection using the RGB image was performed only once, 31 cases were detected as non-fall events even after a fall had been detected from the IMU sensor data. In addition, 11 cases were detected as falls even though no falls had actually occurred. To solve this problem, we performed fall detection using the RGB image ten times and then selected the detection result with the highest probability. As a result, we were able to eliminate misdetection using RGB images, as shown on the right side of Table 7. In addition, we were able to use this double-check method to reduce the number of occurrences of false negatives and false positives from the IMU data to 0, as shown in Table 8. Using the double-check method, we were able to increase the fall detection accuracy achieved with the IMU sensor data from 59.88% to 100.0% in our experimental environment.

VI. CONCLUSION

In this paper, we propose the double-check method using the IMU-L sensor and mobile robot. Since previous studies could not perfectly detect falls by using only an IMU sensor attached on the body, we attempted to overcome this problem by adding a location sensor to the wearable sensor and an RGB sensor on the mobile robot. We adopted the LSTM model to analyze IMU data and selected the DenseNet-121 model to analyze RGB data and achieve the best performance. In IMU-based fall detection, we maximized the precision by using a larger weight for the fall data compared to the non-fall data. Although non-fall activity was classified as a fall by using the IMU data, the mobile robot could move to the site of the fall and confirm the fall with great accuracy. As a result, we achieved perfect performance in detecting a fall in our experimental environment. The proposed method increases the cost and time of falls but could minimize the occurrence of false alarms and maximize the fall detection precision. Our fall detection system can be applied to large spaces for elderly care, such as nursing hospitals and health centers, where robots can easily move. Multiple robots can be deployed to cover each floor in these places. Furthermore, the system may also be installed in a one-story house with elderly occupants. Indoor care robots can widen their roles to address emergency situations by applying our system.

TABLE 8. Confusion Matrices for Proposed Double-Check Fall Detection.

Double-Check Methods using IMU-L and RGB data	Actual Fall	Actual Non-Fall
Predicted Fall	600 (TP)	0 (FP)
Predicted Non-Fall	0 (FN)	1050 (TN)



FIGURE 9. Robot demonstration for fall detection in several places.

VII. FUTURE WORKS

Since we tested our method in a limited laboratory environment, we were able to obtain optimal results. In a less-constrained environment, the precision may be lower. It is expected that follow-up studies will show that considering additional features, such as time or spatial information, will overcome the limitations imposed by real-world environments. Furthermore, the risk of falls may be very low if a person is lying on a bed or sofa or if a person stands up and walks after a fall is detected. If the robot is dispatched to reconfirm the fall even in these less dangerous cases, many unnecessary costs may be incurred. We will improve on results in terms of cost reduction and efficiency by applying a fall detection method that includes this situational information.

ACKNOWLEDGMENT

(Deok-Won Lee and Kooksung Jun contributed equally to this work.)

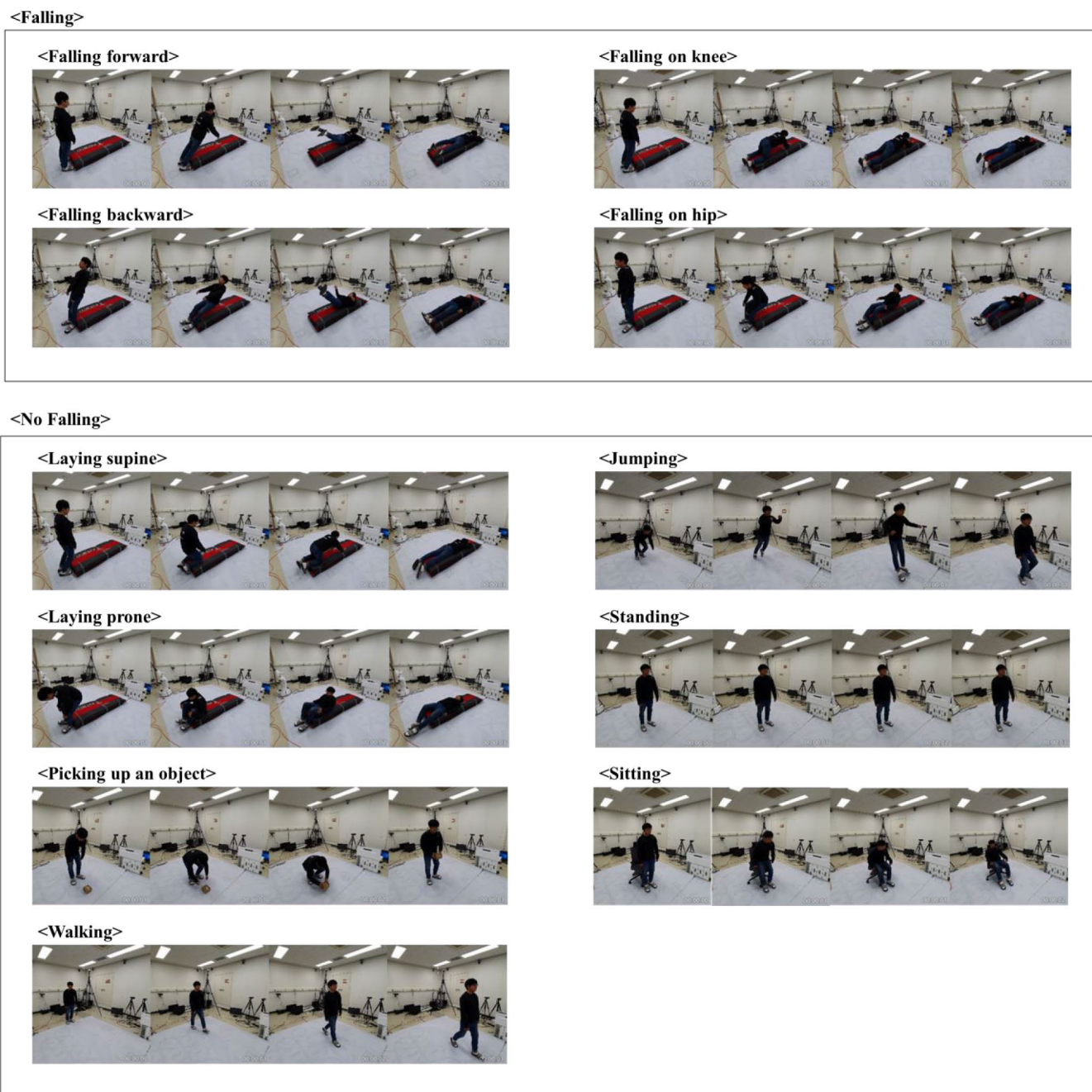


FIGURE 10. Example of IMU-L sensor data acquisition.

APPENDIX A

The proposed fall detecting system has been demonstrated in several places, such as laboratories, aisles, conference rooms, offices, nursing hospitals and health centers. Our system successfully detected human falls in all places, so we verified that the system could work with other environments and backgrounds, which were not trained. Fig. 9 shows the demonstration examples of the proposed fall detection system.

APPENDIX B

The participants in this experiment were five healthy adult males. Data of four types of falls (forward, backward, falling on knee and falling on hip) and seven types of non-falls (laying supine, laying prone, picking up an object, standing, walking, jumping and sitting) were collected. Examples of acquisition experiments and samples of collected IMU-L data are shown in Fig. 10 and Fig. 11, respectively.

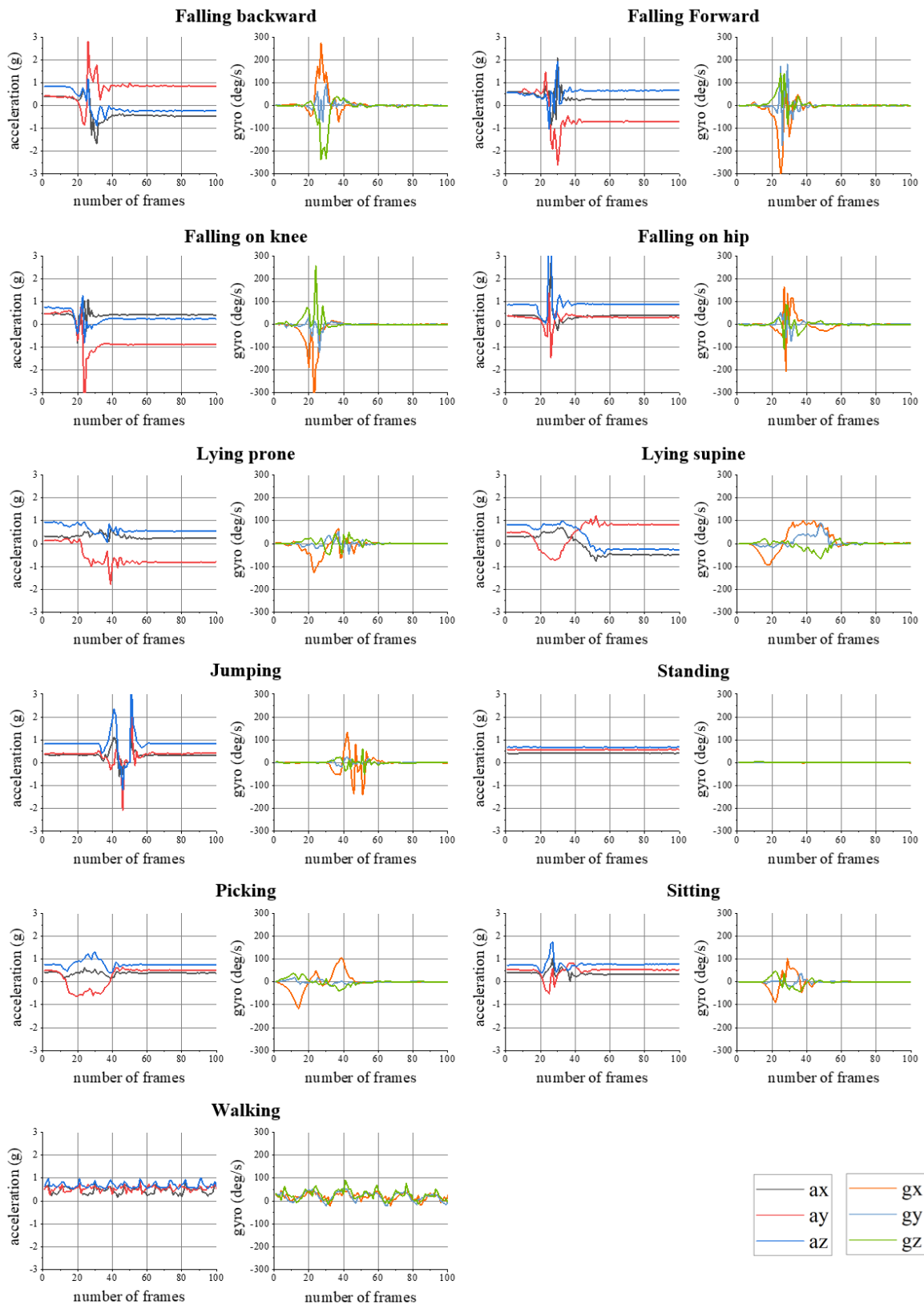


FIGURE 11. Examples of IMU-L sensor data.

REFERENCES

- [1] M. Terroso, N. Rosa, A. T. Marques, and R. Simoes, “Physical consequences of falls in the elderly: A literature review from 1995 to 2010,” *Eur. Rev. Aging Phys. Activity*, vol. 11, no. 1, pp. 51–59, Apr. 2014, doi: [10.1007/s11556-013-0134-8](https://doi.org/10.1007/s11556-013-0134-8).
- [2] World Health Organization. *Falls*. Accessed: May 5, 2020. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/ageing-and-health>
- [3] World Health Organization. *Ageing and Health*. Accessed: May 5, 2020. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/ageing-and-health>
- [4] United Nations Department for Economic and Social Affairs, *World Population Prospects 2019: Highlights*. New York, NY, USA: United Nations Department for Economic and Social Affairs, 2019.
- [5] W. Liu, W. Luo, D. Lian, and S. Gao, “Future frame prediction for anomaly detection—A new baseline,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, New York, NY, USA, Jun. 2018, pp. 6536–6545.
- [6] M. S. Aliakbarian, F. S. Saleh, M. Salzmann, B. Fernando, L. Petersson, and L. Andersson, “Encouraging LSTMs to anticipate actions very early,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Venice, Italy, Oct. 2017, pp. 280–289.
- [7] Y. Wu, L. Zhu, X. Wang, Y. Yang, and F. Wu, “Learning to anticipate egocentric actions by imagination,” *IEEE Trans. Image Process.*, vol. 30, pp. 1143–1152, Dec. 2021, doi: [10.1109/TIP.2020.3040521](https://doi.org/10.1109/TIP.2020.3040521).
- [8] L. Panahi and V. Ghods, “Human fall detection using machine vision techniques on RGB–D images,” *Biomed. Signal Process. Control*, vol. 44, pp. 146–153, Jul. 2018, doi: [10.1016/j.bspc.2018.04.014](https://doi.org/10.1016/j.bspc.2018.04.014).
- [9] E. Cippitelli, S. Gasparri, E. Gambi, S. Spinsante, J. Wahsley, I. Orhany, and T. Lindhy, “Time synchronization and data fusion for RGB–depth cameras and inertial sensors in AAL applications,” in *Proc. IEEE Int. Conf. Commun. Workshop (ICCW)*, London, U.K., Jun. 2015, pp. 265–270.
- [10] E. Cippitelli, S. Gasparri, E. Gambi, and S. Spinsante, “An integrated approach to fall detection and fall risk estimation based on RGB–depth and inertial sensors,” in *Proc. 7th Int. Conf. Softw. Develop. Technol. Enhancing Accessibility Fighting Info-Exclusion*. New York, NY, USA: Association for Computing Machinery, Dec. 2016, pp. 246–253.
- [11] B. Kwolek and M. Kepski, “Fuzzy inference-based fall detection using Kinect and body-worn accelerometer,” *Appl. Soft Comput.*, vol. 40, pp. 305–318, Mar. 2016, doi: [10.1016/j.asoc.2015.11.031](https://doi.org/10.1016/j.asoc.2015.11.031).
- [12] M. Kepski and B. Kwolek, “Detecting human falls with 3-axis accelerometer and depth sensor,” in *Proc. 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Chicago, IL, USA, Aug. 2014, pp. 770–773.
- [13] M. Kepski and B. Kwolek, “Embedded system for fall detection using body-worn accelerometer and depth sensor,” in *Proc. IEEE 8th Int. Conf. Intell. Data Acquisition Adv. Comput. Syst., Technol. Appl. (IDAACS)*, vol. 2, Warsaw, Poland, Sep. 2015, pp. 755–759.
- [14] M. Kepski and B. Kwolek, “Fall detection on embedded platform using Kinect and wireless accelerometer,” in *Proc. Int. Conf. Comput. Handicapped Persons*. Linz, Austria: Springer, Jul. 2012, pp. 407–414.
- [15] M. Kepski and B. Kwolek, “Fall detection using ceiling-mounted 3D depth camera,” in *Proc. Int. Conf. Comput. Vis. Theory Appl. (VISAPP)*, vol. 2, Lisbon, Portugal, Jan. 2014, pp. 640–647.
- [16] O. S. Seredin, A. V. Kopylov, S.-C. Huang, and D. S. Rodionov, “A skeleton features-based fall detection using Microsoft Kinect v2 with one classifier outlier removal,” *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. XLII-2/W12, pp. 189–195, May 2019, doi: [10.5194/isprs-archives-XLII-2-W12-189-2019](https://doi.org/10.5194/isprs-archives-XLII-2-W12-189-2019).
- [17] E. E. Stone and M. Skubic, “Fall detection in homes of older adults using the Microsoft Kinect,” *IEEE J. Biomed. Health Informat.*, vol. 19, no. 1, pp. 290–301, Jan. 2015, doi: [10.1109/JBHI.2014.2312180](https://doi.org/10.1109/JBHI.2014.2312180).
- [18] G. Mastorakis and D. Makris, “Fall detection system using Kinect’s infrared sensor,” *J. Real-Time Image Process.*, vol. 9, no. 4, pp. 635–646, Dec. 2014, doi: [10.1007/s11554-012-0246-9](https://doi.org/10.1007/s11554-012-0246-9).
- [19] S. Gasparri, E. Cippitelli, S. Spinsante, and E. Gambi, “A depth-based fall detection system using a Kinect sensor,” *Sensors*, vol. 14, no. 2, pp. 2756–2775, Feb. 2014, doi: [10.3390/s140202756](https://doi.org/10.3390/s140202756).
- [20] C. Kawatsu, J. Li, and C. J. Chung, “Development of a fall detection system with Microsoft Kinect,” in *Robot Intelligence Technology and Applications*, J. H. Kim, E. Matson, H. Myung, and P. Xu, Eds. Berlin, Germany: Springer, 2013, pp. 623–630.
- [21] A. T. Nghiem, E. Auvinet, and J. Meunier, “Head detection using Kinect camera and its application to fall detection,” in *Proc. 11th Int. Conf. Inf. Sci., Signal Process. Their Appl. (ISSPA)*, Montreal, QC, Canada, Jul. 2012, pp. 164–169.
- [22] Y. Li, K. C. Ho, and M. Popescu, “Efficient source separation algorithms for acoustic fall detection using a Microsoft Kinect,” *IEEE Trans. Biomed. Eng.*, vol. 61, no. 3, pp. 745–755, Mar. 2014, doi: [10.1109/TBME.2013.2288783](https://doi.org/10.1109/TBME.2013.2288783).
- [23] V. Bevilacqua, N. Nuzzolese, D. Barone, M. Pantaleo, M. Suma, D. D’Ambrosio, A. Volpe, C. Loconsole, and F. Stroppa, “Fall detection in indoor environment with Kinect sensor,” in *Proc. IEEE Int. Symp. Innov. Intell. Syst. Appl. (INISTA)*, Alberobello, Italy, Jun. 2014, pp. 319–324.
- [24] Z. A. Mundher and Z. Jiaofei, “A real-time fall detection system in elderly care using mobile robot and Kinect sensor,” *Int. J. Mater., Mech. Manuf.*, vol. 2, no. 2, pp. 133–138, May 2014, doi: [10.7763/IJMMM.2014.V2.115](https://doi.org/10.7763/IJMMM.2014.V2.115).
- [25] T.-T.-T. Tran, T.-L. Le, and J. Morel, “An analysis on human fall detection using skeleton from Microsoft Kinect,” in *Proc. IEEE 5th Int. Conf. Commun. Electron. (ICCE)*, Danang, Vietnam, Jul./Aug. 2014, pp. 484–489.
- [26] C. K. Lee and V. Y. Lee, “Fall detection system based on Kinect sensor using novel detection and posture recognition algorithm,” in *Proc. Int. Conf. Smart Homes Health Telematics*. Singapore: Springer, Jun. 2013, pp. 238–244.
- [27] M. Kepski and B. Kwolek, “Human fall detection using Kinect sensor,” in *Proc. 8th Int. Conf. Comput. Recognit. Syst. (CORES)*. Berlin, Germany: Springer, 2013, pp. 743–752.
- [28] R. Planinc and M. Kampel, “Introducing the use of depth data for fall detection,” *Pers. Ubiquitous Comput.*, vol. 17, no. 6, pp. 1063–1072, Aug. 2013, doi: [10.1007/s00779-012-0552-z](https://doi.org/10.1007/s00779-012-0552-z).
- [29] A. Amini, K. Banitsas, and J. Cosmas, “A comparison between heuristic and machine learning techniques in fall detection using Kinect v2,” in *Proc. IEEE Int. Symp. Med. Meas. Appl. (MeMeA)*, Benevento, Italy, May 2016, pp. 1–6.
- [30] B. Kwolek and M. Kepski, “Human fall detection on embedded platform using depth maps and wireless accelerometer,” *Comput. Methods Programs Biomed.*, vol. 117, no. 3, pp. 489–501, Dec. 2014, doi: [10.1016/j.cmpb.2014.09.005](https://doi.org/10.1016/j.cmpb.2014.09.005).
- [31] A. Buke, F. Gaoli, W. Yongcai, S. Lei, and Y. Zhiqi, “Healthcare algorithms by wearable inertial sensors: A survey,” *China Commun.*, vol. 12, no. 4, pp. 1–12, Apr. 2015, doi: [10.1109/CC.2015.7114054](https://doi.org/10.1109/CC.2015.7114054).
- [32] A. Sorvala, E. Alasaarela, H. Sorvoja, and R. Myllylä, “A two-threshold fall detection algorithm for reducing false alarms,” in *Proc. 6th Int. Symp. Med. Inf. Commun. Technol. (ISMICT)*, La Jolla, CA, USA, Mar. 2012, pp. 1–4.
- [33] A. Hakim, M. S. Huq, S. Shanta, and B. S. K. K. Ibrahim, “Smartphone based data mining for fall detection: Analysis and design,” *Procedia Comput. Sci.*, vol. 105, pp. 46–51, Dec. 2017, doi: [10.1016/j.procs.2017.01.188](https://doi.org/10.1016/j.procs.2017.01.188).
- [34] T. de Quadros, A. E. Lazzaretti, and F. K. Schneider, “A movement decomposition and machine learning-based fall detection system using wrist wearable device,” *IEEE Sensors J.*, vol. 18, no. 12, pp. 5082–5089, Jun. 2018, doi: [10.1109/JSEN.2018.2829815](https://doi.org/10.1109/JSEN.2018.2829815).
- [35] F. De Cillis, F. De Simio, F. Guido, R. A. Incalzi, and R. Setola, “Fall-detection solution for mobile platforms using accelerometer and gyroscope data,” in *Proc. 37th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Milan, Italy, Aug. 2015, pp. 3727–3730.
- [36] F. Wu, H. Zhao, Y. Zhao, and H. Zhong, “Development of a wearable-sensor-based fall detection system,” *Int. J. Telemed. Appl.*, vol. 2015, Feb. 2015, Art. no. 576364, doi: [10.1155/2015/576364](https://doi.org/10.1155/2015/576364).
- [37] S. Madansingh, T. A. Thrasher, C. S. Layne, and B.-C. Lee, “Smartphone based fall detection system,” in *Proc. 15th Int. Conf. Control, Automat. Syst. (ICCAS)*, Ljubljana, Slovenia, Oct. 2015, pp. 370–374.
- [38] X. Hu and X. Qu, “Pre-impact fall detection,” *Biomed. Eng. OnLine*, vol. 15, no. 1, p. 61, Jun. 2016, doi: [10.1186/s12938-016-0194-x](https://doi.org/10.1186/s12938-016-0194-x).
- [39] 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, doi: [10.1109/JSEN.2008.2012212](https://doi.org/10.1109/JSEN.2008.2012212).
- [40] P. Pierleoni, A. Belli, L. Palma, M. Pellegrini, L. Pardini, and S. Valenti, “A high reliability wearable device for elderly fall detection,” *IEEE Sensors J.*, vol. 15, no. 8, pp. 4544–4553, Aug. 2015, doi: [10.1109/JSEN.2015.2423562](https://doi.org/10.1109/JSEN.2015.2423562).
- [41] G. Sannino, I. De Falco, and G. De Pietro, “A supervised approach to automatically extract a set of rules to support fall detection in an mHealth system,” *Appl. Soft Comput.*, vol. 34, pp. 205–216, Sep. 2015, doi: [10.1016/j.asoc.2015.04.060](https://doi.org/10.1016/j.asoc.2015.04.060).

- [42] B. Najafi, K. Aminian, A. Paraschiv-Ionescu, F. Loew, C. J. Bula, and P. Robert, "Ambulatory system for human motion analysis using a kinematic sensor: Monitoring of daily physical activity in the elderly," *IEEE Trans. Biomed. Eng.*, vol. 50, no. 6, pp. 711–723, Jun. 2003, doi: [10.1109/TBME.2003.812189](https://doi.org/10.1109/TBME.2003.812189).
- [43] A. Alarif and A. Alwadain, "Killer heuristic optimized convolution neural network-based fall detection with wearable IoT sensor devices," *Measurement*, vol. 167, Jan. 2021, Art. no. 108258, doi: [10.1016/j.measurement.2020.108258](https://doi.org/10.1016/j.measurement.2020.108258).
- [44] J. A. B. Madrigal, J. C. Negrete, R. M. Guerrero, L. A. C. Rodríguez, and H. Sossa, "3D motion tracking of the shoulder joint with respect to the thorax using MARG sensors and data fusion algorithm," *Bio-cybernet. Biomed. Eng.*, vol. 40, no. 3, pp. 1205–1224, Jul. 2020, doi: [10.1016/j.bbe.2020.04.008](https://doi.org/10.1016/j.bbe.2020.04.008).
- [45] C. Gil, H. Calvo, and H. Sossa, "Learning an efficient gait cycle of a biped robot based on reinforcement learning and artificial neural networks," *Appl. Sci.*, vol. 9, no. 3, p. 502, Feb. 2019, doi: [10.3390/app9030502](https://doi.org/10.3390/app9030502).
- [46] B. Koo, J. Kim, T. Kim, H. Jung, Y. Nam, and Y. Kim, "Post-fall detection using ANN based on ranking algorithms," *Int. J. Precis. Eng. Manuf.*, vol. 21, no. 10, pp. 1985–1995, Aug. 2020, doi: [10.1007/s12541-020-00398-6](https://doi.org/10.1007/s12541-020-00398-6).
- [47] T. Mauldin, M. Canby, V. Metsis, A. Ngu, and C. Rivera, "SmartFall: A smartwatch-based fall detection system using deep learning," *Sensors*, vol. 18, no. 10, p. 3363, Oct. 2018.
- [48] S. Q. Mahdi, S. K. Gharghan, and M. A. Hasan, "FPGA-based neural network for accurate distance estimation of elderly falls using WSN in an indoor environment," *Measurement*, vol. 167, Jan. 2021, Art. no. 108276, doi: [10.1016/j.measurement.2020.108276](https://doi.org/10.1016/j.measurement.2020.108276).
- [49] S. Porta, A. Martínez, N. Millor, M. Gómez, and M. Izquierdo, "Relevance of sex, age and gait kinematics when predicting fall-risk and mortality in older adults," *J. Biomech.*, vol. 105, May 2020, Art. no. 109723, doi: [10.1016/j.jbiomech.2020.109723](https://doi.org/10.1016/j.jbiomech.2020.109723).
- [50] N. El-Bendary, Q. Tan, F. C. Pivot, and A. Lam, "Fall detection and prevention for the elderly: A review of trends and challenges," *Int. J. Smart Sens. Intell. Syst.*, vol. 6, no. 3, pp. 1230–1266, Jun. 2013, doi: [10.21307/ijssis-2017-588](https://doi.org/10.21307/ijssis-2017-588).
- [51] L. P. Malasinghe, N. Ramzan, and K. Dahal, "Remote patient monitoring: A comprehensive study," *J. Ambient Intell. Hum. Comput.*, vol. 10, no. 1, pp. 57–76, Jan. 2019, doi: [10.1007/s12652-017-0598-x](https://doi.org/10.1007/s12652-017-0598-x).
- [52] Z. Wang, V. Ramamoorthy, U. Gal, and A. Guez, "Possible life saver: A review on human fall detection technology," *Robotics*, vol. 9, no. 3, p. 55, Jul. 2020, doi: [10.3390/robotics9030055](https://doi.org/10.3390/robotics9030055).
- [53] Y. Nizam, M. N. H. Mohd, and M. M. A. Jamil, "A study on human fall detection systems: Daily activity classification and sensing techniques," *Int. J. Integr. Eng.*, vol. 8, no. 1, pp. 35–43, Nov. 2016.
- [54] L. Martínez-Villaseñor, H. Ponce, J. Brieva, E. Moya-Albor, J. Núñez-Martínez, and C. Peñafor-Asturiano, "UP-fall detection dataset: A multimodal approach," *Sensors*, vol. 19, no. 9, p. 1988, Apr. 2019, doi: [10.3390/s19091988](https://doi.org/10.3390/s19091988).
- [55] S. W. Pienaar and R. Malekian, "Human activity recognition using LSTM-RNN deep neural network architecture," in *Proc. IEEE 2nd Wireless Afr. Conf. (WAC)*, Pretoria, South Africa, Aug. 2019, pp. 1–5.
- [56] M. Qi, Y. Wang, J. Qin, A. Li, J. Luo, and L. Van Gool, "StagNet: An attentive semantic RNN for group activity and individual action recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 2, pp. 549–565, Feb. 2020, doi: [10.1109/TCSVT.2019.2894161](https://doi.org/10.1109/TCSVT.2019.2894161).
- [57] S. Shin and W.-Y. Kim, "Skeleton-based dynamic hand gesture recognition using a part-based GRU-RNN for gesture-based interface," *IEEE Access*, vol. 8, pp. 50236–50243, Mar. 2020, doi: [10.1109/ACCESS.2020.2980128](https://doi.org/10.1109/ACCESS.2020.2980128).
- [58] K. Jun, D.-W. Lee, K. Lee, S. Lee, and M. S. Kim, "Feature extraction using an RNN autoencoder for skeleton-based abnormal gait recognition," *IEEE Access*, vol. 8, pp. 19196–19207, Jan. 2020, doi: [10.1109/ACCESS.2020.2967845](https://doi.org/10.1109/ACCESS.2020.2967845).
- [59] D.-W. Lee, K. Jun, S. Lee, J.-K. Ko, and M. S. Kim, "Abnormal gait recognition using 3D joint information of multiple Kinects system and RNN-LSTM," in *Proc. 41st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Berlin, Germany, Jul. 2019, pp. 542–545.
- [60] M. M. Hasan and H. A. Mustafa, "Multi-level feature fusion for robust pose-based gait recognition using RNN," *Int. J. Comput. Sci. Inform. Secur.*, vol. 18, no. 1, pp. 20–31, Jan. 2020.
- [61] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of CNN and RNN for natural language processing," 2017, *arXiv:1702.01923*. [Online]. Available: <http://arxiv.org/abs/1702.01923>
- [62] V.-K. Tran and L.-M. Nguyen, "Natural language generation for spoken dialogue system using RNN encoder-decoder networks," 2017, *arXiv:1706.00139*. [Online]. Available: <http://arxiv.org/abs/1706.00139>
- [63] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inform. Process. Syst.*, New York, NY, USA, 2015, pp. 91–99.
- [64] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 6154–6162.
- [65] S. Gidaris and N. Komodakis, "Object detection via a multi-region and semantic segmentation-aware CNN model," in *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, Dec. 2015, pp. 1134–1142.
- [66] Y. Cao, Y. Chen, and D. Khosla, "Spiking deep convolutional neural networks for energy-efficient object recognition," *Int. J. Comput. Vis.*, vol. 113, no. 1, pp. 54–66, May 2015, doi: [10.1007/s11263-014-0788-3](https://doi.org/10.1007/s11263-014-0788-3).
- [67] M. Schwarz, H. Schulz, and S. Behnke, "RGB-D object recognition and pose estimation based on pre-trained convolutional neural network features," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Seattle, WA, USA, May 2015, pp. 1329–1335.
- [68] J. M. Gandarias, A. J. Garcia-Cerezo, and J. M. Gomez-de-Gabriel, "CNN-based methods for object recognition with high-resolution tactile sensors," *IEEE Sensors J.*, vol. 19, no. 16, pp. 6872–6882, Aug. 2019, doi: [10.1109/JSEN.2019.2912968](https://doi.org/10.1109/JSEN.2019.2912968).
- [69] Stanford Education. *Transfer Learning*. Accessed: May 5, 2020. [Online]. Available: <https://cs231n.github.io/transfer-learning/>
- [70] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735).
- [71] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. 15th Annu. Conf. Int. Speech Commun. Assoc.*, Singapore, Sep. 2014, pp. 1–5.
- [72] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*. [Online]. Available: <http://arxiv.org/abs/1406.1078>
- [73] J. Chung, G. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*. [Online]. Available: <http://arxiv.org/abs/1412.3555>
- [74] ImageNet. *Large Scale Visual Recognition Challenge (ILSVRC)*. Accessed: May 5, 2020. [Online]. Available: <http://www.image-net.org/challenges/LSVRC/>
- [75] Apple. *Use Fall Detection With Apple Watch*. Accessed: Feb. 14, 2021. [Online]. Available: <https://support.apple.com/en-us/HT208944>
- [76] Samsung. *Setting Up Fall Detection Function on the Galaxy Watch3*. Accessed: Feb. 14, 2021. [Online]. Available: <https://www.samsung.com/au/support/mobile-devices/set-up-detect-falls/>
- [77] 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, doi: [10.3390/s17071513](https://doi.org/10.3390/s17071513).
- [78] L. Blond and F. Olesen, "Unpacking the cultural baggage of travelling robots," in *Designing Robots, Designing Humans*, C. Hasse and D. M. Søndergaard, Eds. New York, NY, USA: Routledge, 2019, pp. 111–131.
- [79] L. Blond, "Studying robots outside the lab: HRI as ethnography," *J. Paladyn Behav. Robot.*, vol. 10, no. 1, pp. 117–127, Mar. 2019, doi: [10.1515/pjbr-2019-0007](https://doi.org/10.1515/pjbr-2019-0007).
- [80] Robocare. *Silbot*. Accessed: May 5, 2020. [Online]. Available: http://robocare.co.kr/pages/product03_en.php
- [81] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: An open-source robot operating system," in *Proc. ICRA Workshop Open Source Softw.*, Kobe, Japan, 2009, vol. 3, no. 3.2, p. 5.
- [82] A. Koubâa, *Robot Operating System (ROS): The Complete Reference*, vol. 2. Cham, Switzerland: Springer, 2017.
- [83] ROS. *ROS*. Accessed: May 5, 2020. [Online]. Available: <https://www.ros.org/>
- [84] A. S. Kundu, O. Mazumder, A. Dhar, and S. Bhaumik, "Occupancy grid map generation using 360° scanning lidar for indoor mobile robot navigation," in *Proc. IEEE 1st Int. Conf. Control, Meas. Instrum. (CMI)*, Kolkata, India, Jan. 2016, pp. 464–468.
- [85] ASUS. *Use Xtion PRO Developer Solution to Make Motion-Sensing Applications and Games*. Accessed: May 5, 2020. [Online]. Available: https://www.asus.com/3D-Sensor/Xtion_PRO_LIVE

[86] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994, doi: [10.1109/72.279181](https://doi.org/10.1109/72.279181).

[87] Torch Contributor. *Torchvision Models*. Accessed: Dec. 2, 2020. [Online]. Available: <https://pytorch.org/docs/stable/torchvision/models.html>

[88] Yrevar. *ImageNet 1000 Class Index to Labels*. Accessed: Dec. 2, 2020. [Online]. Available: <https://gist.github.com/yrevar/942d3a0ac09ec9e5eb3a>

[89] H. L. Bank and M. K. Schmehl, “Parameters for evaluation of viability assays: Accuracy, precision, specificity, sensitivity, and standardization,” *Cryobiology*, vol. 26, no. 3, pp. 203–211, Jun. 1989, doi: [10.1016/0011-2240\(89\)90015-1](https://doi.org/10.1016/0011-2240(89)90015-1).

[90] A. R. Midgley, G. D. Niswender, and R. W. Rebar, “Principles for the assessment of the reliability of radioimmunoassay methods (precision, accuracy, sensitivity, specificity),” *Eur. J. Endocrinol.*, vol. 62, no. 1, pp. 163–184, Feb. 1969, doi: [10.1530/acta.0.062s163](https://doi.org/10.1530/acta.0.062s163).

[91] Wikipedia. *Precision and Recall*. Accessed: Feb. 8, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Precision_and_recall



DEOK-WON LEE received the B.S. degree in electronics and electric wave and information engineering from Chungnam National University, in 2009, and the M.S. degree in electrical and electronics engineering from Yonsei University in 2013. He is currently pursuing the Ph.D. degree with the Gwangju Institute of Science and Technology. His current research interests include healthcare robotics, artificial intelligence, and pattern recognition.



KOOKSUNG JUN received the B.S. degree in mechanical engineering and the M.S. degree in intelligent robotics from the Gwangju Institute of Science and Technology, Gwangju, South Korea, in 2018 and 2019, respectively, where he is currently pursuing the Ph.D. degree. His current research interests include healthcare robotics, artificial intelligence, and pattern recognition.



KHAWAR NAHEEM received the B.S. degree in electrical engineering from COMSATS University Islamabad, Pakistan, in 2011, and the M.S. degree in distributed generation systems and control from Dongguk University, Seoul, South Korea, in 2014. He is currently pursuing the Ph.D. degree with the Gwangju Institute of Science and Technology. His current research interests include UWB-based multi-agent indoor localization and navigation systems, lighter-than-air autonomous robots, human–robot interaction, and sensor fusion.



MUN SANG KIM (Member, IEEE) received the B.S. and M.S. degrees in mechanical engineering from Seoul National University, Seoul, South Korea, in 1980 and 1982, respectively, and the Dr.-Ing. degree in robotics from the Technical University of Berlin, Berlin, Germany in 1987. From 1987 to 2016, he was a Research Scientist with the Korea Institute of Science and Technology, Seoul. He led the Advanced Robotics Research Center, in 2000, and became the Director of the Intelligent Robot—The Frontier 21 Program, in October 2003, one of the most challenging research programs in South Korea. He is currently a Professor with the School of Integrated Technology, Gwangju Institute of Science and Technology. His current research interests include healthcare robotics, UWB-based indoor localization systems, and culture technology.

...