International Journal of Internet, Broadcasting and Communication Vol.12 No.2 120-126 (2020)
http://dx.doi.org/10.7236/IJIBC.2020.12.2.120

IJIBC 20-2-16

Design of Falling Recognition Application System using Deep Learning

TaeWoo Kwon[1], Jong-Yong Lee[2] and Kye-Dong Jung[2]

[1]*Master, Department of Information System KwangWoon University Graduate School of Smart Convergence, Seoul 01897, Korea*
[2]*Professor, Ingenium College of liberal arts, KwangWoon University, Seoul 01897, Korea*

e-mail : {Ryuk9302, jyonglee, gdchung }@kw.ac.kr

Abstract

Studies are being conducted regarding falling recognition using sensors on smartphones to recognize falling in human daily life. These studies use a number of sensors, mostly acceleration sensors, gyro sensors, motion sensors, etc. Falling recognition system processes the values of sensor data by using a falling recognition algorithm and classifies behavior based on thresholds. If the threshold is ambiguous, the accuracy will be reduced. To solve this problem, Deep learning was introduced in the behavioral recognition system. Deep learning is a kind of machine learning technique that computers process and categorize input data rather than processing it by man-made algorithms. Thus, in this paper, we propose a falling recognition application system using deep learning based on smartphones. The proposed system is powered by apps on smartphones. It also consists of three layers and uses DataBase as a Service (DBaaS) to handle big data and address data heterogeneity. The proposed system uses deep learning to recognize the user's behavior, it can expect higher accuracy compared to the system in the general rule base.

Keywords: *Deep Learning, Smart Phone, Application, Falling Recognition System, DBaaS.*

1. Introduction

With The falling is not only at a high risk of injury, but it can also lead to secondary injuries that occur thereafter. Because of this, various studies are being conducted to recognize falling [1][2]. Most studies use various sensors to collect sensor data from sensors that are measured when falling and classify them using a falling recognition algorithm. Recently, sensors built into smartphones are being used due to the miniaturization of sensors [4][5]. Major technologies that collect sensor data using acceleration sensors, gyro sensors, motion sensors, and others are used for falling recognition. While most human behaviors are the same, in the case of falling, the definition of the beginning and end of the act is ambiguous. This is also true of sensor data. For this reason, the majority of studies use sensor data thresholds to recognize falling. However, how thresholds are used is also less accurate because they are perceived only in terms of thresholds regardless of the data pattern. Research has been on the rise to introduce AI-based deep learning into falling recognition

Manuscript Received: April. 10, 2020 / Revised: April. 16, 2020 / Accepted: April. 22, 2020
Corresponding Author: gdchung@kw.ac.kr
Tel: +82-2-940-5288
Professor, Ingenium College of liberal arts, Kwangwoon University, Seoul, Korea

technology [7][8].

Deep learning based on artificial intelligence uses a network called artificial neural network to classify data. Deep learning uses artificial neural networks to set the optimum weight for classification by non-human computers. The process of updating weights is called the "train", and the weights produced using the artificial neural network are updated using error backpropagation. The performance of artificial intelligence technologies, including deep learning, is high in the amount of data and the more varied In other words, high performance can be expected only when big data is used as input, and for this, the system needs a processing unit for big data processing. In addition, pre-processing of input data is required to use deep learning. Pre-processing involves creating a Ground Truth (GT) by entering labels into some data for processing and learning of sensor data. In this paper, we propose a smartphone application system that collects sensor data based on smartphones and recognizes falling by using deep learning on servers. DBaaS is used to handle big data and solve data heterogeneity [9]. The sensors used for acquisition use acceleration sensors and gyro sensors and the pre-processing of sensor data uses the Signal Vector Magnitude (SVM) and the Rotation Angle algorithm. SVMs are useful values for recognizing behavior with movements such as falling and running. And the angle of rotation is a useful value for recognizing static behavior, such as standing. In addition, users can enter labels directly when collecting sensor data to handle GT.

The proposed system collects sensor data from smartphones, pre-processed it on servers and learns it using artificial neural networks. In addition, server overload and latency problems that occur when processing on a single server are divided into three layers. Each layer consists of the User Layer, the Data Management Layer, and the Data Processing Layer. The User Layer corresponds to a smartphone and consists of an application page for collecting and transmitting sensor data and verifying the results. The Data Management Layer consists of modules for data mapping for managing data and resolving heterogeneity. The Data Processing Layer consists of the pre-processed of sensor data and the part of the model that trains using an artificial neural network.

The composition of this paper is as follows. Chapter 2 describes behavioral awareness skills, deep learning as related works. Chapter 3 describes the architecture and description of each layer of the proposed system and describes the behavior and flow of the system. Chapter 4 describes apps that are implemented based on Smartphones. Chapter 5 summarizes this paper in the conclusion

2. Related Works

2.1 Behavior Recognition Technique

Behavior recognition techniques refer to Activity of Daily Living (ADL) [1][2]. Most of the behavior recognition techniques use a variety of sensors. The sensor data collected is converted according to the algorithm and behavior is classified by the threshold of these values [3]. The main algorithm used in behavior recognition is the SVM [4][5]. The SVM can be obtained using acceleration sensor data. Equation 1 uses acceleration sensor data to obtain SVM values. In addition, the SVM value is divided by the gravitational acceleration of , showing without movement as 1G.

$$SVM(g) = \frac{\sqrt{(Acc_x)^2+(Acc_y)^2+(Acc_z)^2}}{9.8} \tag{1}$$

Dynamic behavior is easy to apply to the SVM because of the large variation in acceleration sensor data.

However, for the static state, there is little change in the acceleration sensor data, making it difficult to recognize the behavior. In addition to the SVM value, a method was proposed to obtain the rotation angle of a smartphone using a spherical coordinate system to recognize static state [5].

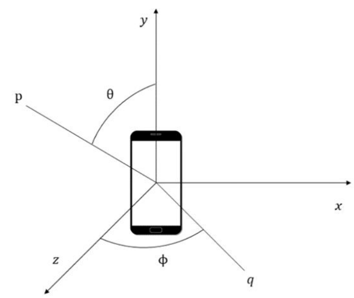


Figure 1. Rotation angle on smartphone

In Figure 1, Theta(θ) algorithm shows the proposed rotation angle and corresponds to θ. Equation 2 is for obtaining the angle of rotation, and θ has a radian value, so convert it to a degree value.

$$\theta_{radian} = \cos^{-1}\left(\frac{Acc_y}{\sqrt{Acc_x{}^2+Acc_y{}^2+Acc_z{}^2}}\right) \times \frac{180}{\pi} \tag{2}$$

2.2 Deep Learning

Deep learning is an area of artificial intelligence, and it is a technology in which computers train people's minds to find the optimum weight [6][7]. Deep learning trains using a network called an artificial neural network. The artificial neural network has a structure consisting of several layers of hidden layers. The values entered in the input layer are weighted by the activation function set in the hidden layer of the different stages. This weight changes using the hidden layer, and the output layer categorizes the data based on the values from the last hidden layer. The types of functions are stair functions, sigmoid functions, reLU functions, hyperbolic tangent, etc. Deep learning consists of training data, GT data, and test data. Training data is the data used in actual training. This means the data used in training and the GT data. The GT data corresponds to the label data at the time of training. Test data means untrained data. The existence of GT data is divided into supervised learning and unsupervised learning. In other words, GT data is categorized as supervised learning, if not, unsupervised learning. Recent deep learning has been attempted in various fields. Especially in image processing, CNN is the main study [10]. A study on reinforcement learning used in games such as AlphaGo is also a major study.

3. Design of Falling Recognition Application System Using Deep Learning

3.1 Deep Learning Model

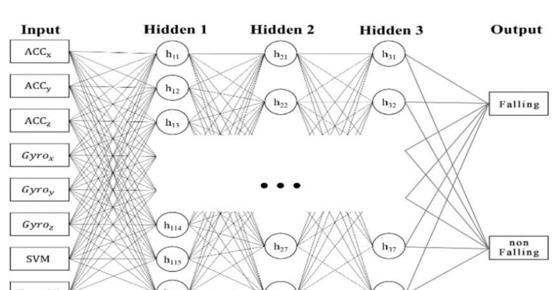


Figure 2. Designed neural networks

This section describes the Deep Learning Model used in the proposed system. The proposed system collects data from acceleration sensors, gyro sensors and GPS sensors from smartphones. The sensor data collected is pre-processed and used. For pre-processed algorithms used, use the SVM, theta algorithm described in Chapter 2. In addition, the data is divided by the maximum value of each sensor to normalize the data. GT normalizes the label value through one-hot encoding. To recognize fall, a deep running model is constructed as shown in Figure 2. There are three hidden layers, and the number of nodes for each layer was set to 16, 8, and 8. The INPUT has data from the acceleration sensor and gyro sensor, normalized by pre-processed, and eight values from the SVM and the rotation angle. OUTPUT has two values: "Falling" and "non Falling". The activation function used for each layer was used to efficiently calculate the reLU. The output layer was selected based on probability values corresponding to each OUPUT using the softmax function.

3. System Architecture & Components

The proposed system consists of three layers. Each layer consists of the User Layer, the Data Management Layer, and the Data Processing Layer. User Layer is composed of applications for measuring and transmitting sensor data by utilizing sensors such as Smartphones. In addition, the collected sensor data and learning data can check the results. The Data Management Layer consists of modules for collecting sensor data, resolving data heterogeneity, and processing data flow. Data Management Layer uses DBaaS and provides data management services such as data mapping. The Data Processing Layer consists of a data pre-processing module for using artificial neural networks and a module for creating models by learning using deep learning and a module for managing and providing learned models. Figure 3 shows the architecture of the proposed system.

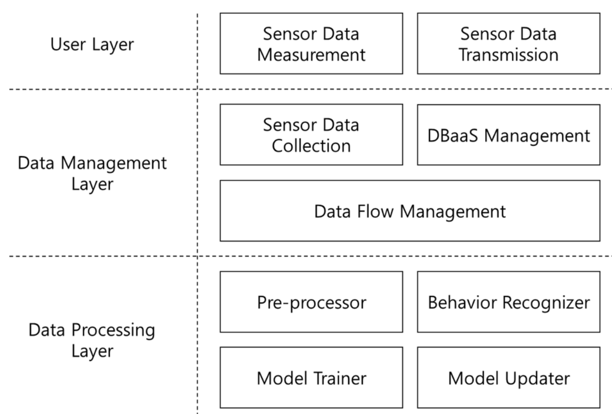


Figure 3. Architecture of proposed system

User Layer shows the Sensor Data Measurement and Sensor Data Transmission as components of the User Layer. Sensor Data Measurement serves to measure sensor data from applications on smartphones, and the proposed system stores acceleration sensor, gyro sensor and GPS sensor data in JSON format. Sensor Data Transmission sends measured sensor data to the Data Management Layer through Sensor Data Measurement. At this time, the user's smartphone ID and time information are sent together.

Data Management Layer shows the components of the Data Management Layer. Components include Sensor Data Collection, DBaaS Management, and Data Flow Management. Sensor Data Collection serves to collect sensor data sent from the user layer. Messages are sent to Data Flow Management and DBaaS Management to store JSON data in the DB. To address data heterogeneity, DBaaS Management provides mapping services that manage mapping information, such as a standard schema for sensor data, and map sensor data to the DB form. Data Flow Management controls the overall flow of data. In other words, it manages storage of sensor data, request of mapping data, transmission of sensor data to pre-processors, and deep learning models.

Data Processing Layer shows the components of the Data Processing Layer. The Data Processing Layer consists of a Pre-processor, a Behavioral Recognizer, a Model Trainer, and a Model Updater. Pre-processors are data pre-processors that verify that the axis on the smartphone is rotated and convert the axis so that the z-axis is up and down. The value of sensor data is then processed through the SVM and the rotation angle algorithm. In addition, since the precision (range) of the sensors varies from sensor to sensor, the sensor is divided by the maximum value of the sensor and the data is normalized to be expressed as a value between 0 and 1. Normalize GT through one-hot encoding. Behavior Recognizer recognizes behavior by inputting sensor data. Recognize behavior when there is a pre-trained model. If there are no trained models, send a message to the Model Trainer to learn the models. Model Trainers use pre-processed sensor data as input values to learn the deep learning model. Parameters to be used should be set in advance and can be modified by developers. Model Updater helps users update their models by receiving feedback when they mis-predict their behavior. Updating a model is done by loading a previously learned model and updating its weight.

3.3 System Operation and Flow

This section describes the operation and flow of the proposed system. Figure 4 shows the flow chart to the pre-processing process of sensor data. ① is a process of measuring sensor data through Smartphone application and saving it as a JSON file. ② is the process of sending saved JSON files to Sensor Data Collection in the Data Management Layer. ③ is the process of parsing and sending the received JSON file to Data Flow Management. ④ is the process of requesting and responding to DBaaS Management for standard schema information from sensors. This process maps sensor data to the standard format used for the database. ⑤ is the process of sending to pre-processors for pre-processing of sensor data and re-transmitting pre-processed data. ⑥ is the process of storing pre-processed data in the DB.

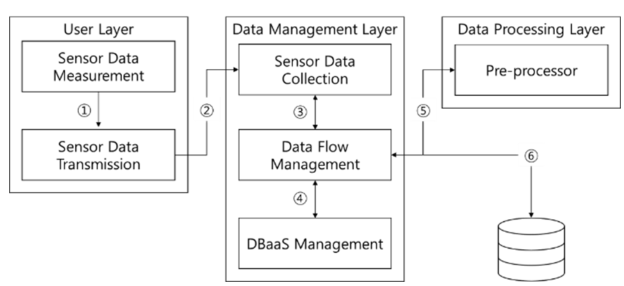


Figure 4. System Flow I – until pre-processing.

Figure 5 shows a flow chart in the absence of a trained deep learning model. ① is the process of requesting and responding to data stored in the DB for learning the model. The requested data are the acceleration sensor and gyro sensor data, the SVM, the rotation angle data, and the GT data. Acceleration sensors, gyro sensors, SVMs, and rotation angle data are then used as input values for behavioral awareness, and GT data as label data. ② is the process of sending data from the DB after verifying that there are no pre-trained models. ③ is the process of training the deep learning model, with the extension ".hdf5". ④ sends the file names of the generated models and details of the models to Data Flow Management.

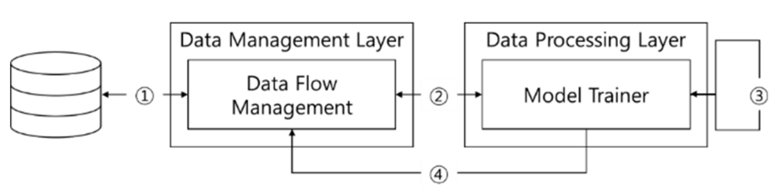


Figure 5. System Flow II – model train.

Figure 6 shows the process of updating the model if the behavior is recognized and misrecognized. The process of collecting and pre-processing sensor data is as shown in Figure 4, and is omitted. ① is the process of transferring pre-processed data to the Behavior Recognizer. ② is the process of requesting and responding to a pre-trained model. ③ is the process of transmitting the results of a recognized behavior. ④ is the process of checking results to users and inputting labels to feedback them so that they can retrain. ⑤ is the process of requesting retraining from Model Updater and sending the data and GT if there is feedback. ⑥ is the process of retraining through pre-trained models and added data. ⑦ is the process of transferring the file name and information of the updated model.

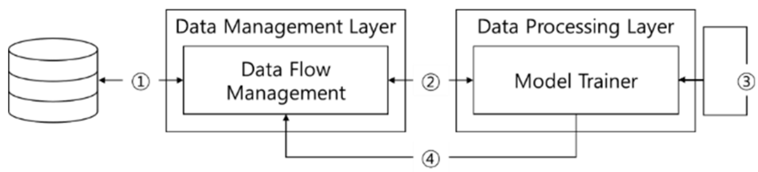
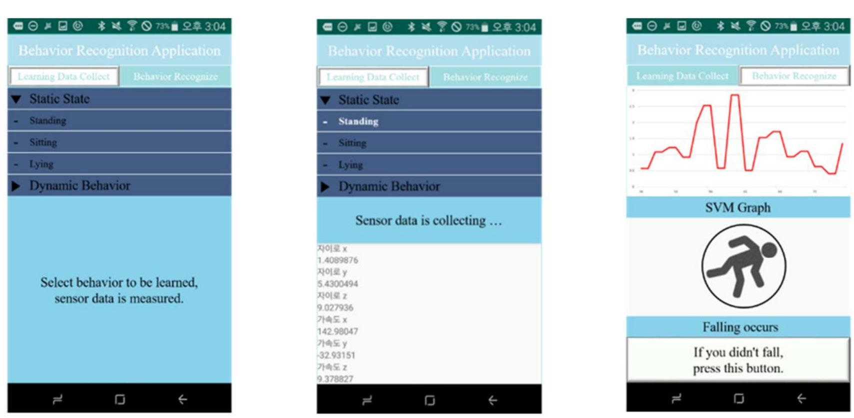


Figure 6. System Flow III – model update.

4. Applying of System

Figure 7 (a) is the screen when the Learning Data Collect function is executed. As can be seen in Figure 7, it is divided into Static State and Dynamic Behavior, where the user labels each behavior to generate a GT. When GT is generated, the corresponding behavior of the label shall be taken for the measurement time corresponding to the label. Figure 7 (b) is the screen when the Standing of Static State is selected. When a user selects a label, the application displays the value of the smartphone sensor in the screens of the middle and measures the data for a period of time (about 3,000 ms). Once the measurement of the data is complete, the data is stored in the raspberry pie's database via data communication such as Wi-Fi. The data stored are the values of the x, y, and z-axis of the gyro sensor and the values of the x, y, z-axis of the acceleration sensor. Figure 7 (c) is the result screen when the Behavior Recognize function is implemented that recognizes behavior based on trained models and sensor data. The result screen can display the SVM graph and the Rotation Angle graph, and two graphs can be viewed in the overlap. It also recognizes the actions taken by the user and outputs icons accordingly. If a falling occurs, check whether or not a falling occurs. If it is not a falling but is output to a falling, it is collected as data for re-learning the Deep Learning model by pressing the button for feedback.



(a) Learning Data Collect Function Screen.

(b) Screen of Selecting Behavior.

(c) Result Screen.

Figure 7. Application screen

5. Conclusion

In this paper, we propose the design of a falling recognition application system using smartphone-based deep learning. In addition, sensor data was measured and collected through smartphone applications and learned through artificial neural networks so that the results recognized as falling were printed on the application. For falling recognition, deep learning was used rather than classified according to the threshold of values calculated by the algorithms used previously. This is also applicable for thresholds that are clear and ambiguous. The proposed system has been configured on a DBaaS basis for efficient processing of sensor data, which is also effective in resolving data heterogeneity. The system was also divided into three layers to minimize the waiting time and server overload when implemented as a single layer. The proposed system may be introduced in the falling notification system at a later date. This could minimize secondary injuries by informing medical officials and caregivers of falling in the elderly or patients at hospitals. In the future, studies are needed that can be classified using images, not just by using acceleration sensor data. This can be used for cross-validation and improved accuracy.

References

[1] Kim, N. H., and Yu, Y. S. (2013). Fall recognition algorithm using gravity-weighted 3-axis accelerometer data. *Journal of the Institute of Electronics and Information Engineers*, 50(6), 254-259. DOI: https://doi.org/10.5573/ieek.2013.50.6.254

[2] Su, X., Tong, H., and Ji, P. (2014). Activity recognition with smartphone sensors. *Tsinghua science and technology*, 19(3), 235-249. DOI: https://doi.org/10.1109/TST.2014.6838194

[3] Özdemir, A., and Barshan, B. (2014). Detecting falls with wearable sensors using machine learning techniques. Sensors, 14(6), 10691-10708. DOI: https://doi.org/10.3390/s140610691

[4] Kwon, T.W, Lee, J. Y., and Jung, K. D. (2017). Design of Cloud-based Context-aware System Based on Falling Type. *International Journal of Internet, Broadcasting and Communication*, 9(4), 44-50. IJIBC. DOI: http://dx.doi.org/10.7236/IJIBC.2018.10.3.42

[5] Kwon, T.W, Yun, D.Y, Lee, J.Y and Jung, K.D. (2018). A Study of Behaviors Recognition Method using Smartphone Sensors. *Journal of Engineering and Applied Sciences*, (13): 8722-8725. DOI: https://doi.org/10.36478/jeasci.2018.8722.8725

[6] Batista, G. E., Prati, R. C., and Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1), 20-29. DOI: https://doi.org/10.1145/1007730.1007735

[7] Rieck, K., Trinius, P., Willems, C., and Holz, T. (2011). Automatic analysis of malware behavior using machine learning. *Journal of Computer Security*, 19(4), 639-668. DOI: https://doi.org/10.3233/JCS-2010-0410

[8] Shin, S., and Cha, J. (2018). Human Activity Recognition System Using Multimodal Sensor and Deep Learning Based on LSTM. *Transactions of the Korean Society of Mechanical Engineers*. A, 42(2), 111-121. DOI: https://doi.org/10.3795/KSME-A.2018.42.2.111

[9] Lehner, W., and Sattler, K. U. (2010, March). Database as a service (DBaaS). *In 2010 IEEE 26th International Conference on Data Engineering* (ICDE 2010) (pp. 1216-1217). IEEE. DOI: https://doi.org/10.1109/ICDE.2010.5447723

[10] Lee, S., Kim, H., Seok, H., & Nang, J. (2017). Comparison of Fine-Tuned Convolutional Neural Networks for Clipart Style Classification. *International Journal of Internet, Broadcasting and Communication*, 1(1), 4. IJIBC. DOI: https://doi.org/10.7236/IJIBC.2017.9.4.1