Fall Detection of Elderly in Ambient Assisted Smart Living Using CNN Based Ensemble Approach

Sakshi Shukralia
Department of Computer Science and Engineering
Netaji Subhas University of Technology
New Delhi, India
sakshishukralia12@gmail.com
[0000-0001-9313-9308]

M.P.S Bhatia
Department of Computer Science and Engineering
Netaji Subhas University of Technology
New Delhi, India
mpsbhatia@nsut.ac.in
[0000-0001-7190-9770]

Pinaki Chakraborty
Department of Computer Science and Engineering
Netaji Subhas University of Technology
New Delhi, India
pinaki_chakraborty_163@yahoo.com
[0000-0002-2010-8022]

Abstract—In recent years, there has been a significant increase in ambient assisted living and smart environment homes that utilize a range of technologies to enhance the quality of life for elderly people. Fall detection is an essential service that smart home healthcare can provide as falls can pose a significant threat to the independence and health of individuals over 65 years old. This article introduces the ISBFD (Inertial Sensor Based Fall Detection) concept, which aims to identify elderly persons who fall and alert family members or carers right away. The proposed model employs data from the accelerometer sensor of a smartphone in real-time. This data is then processed by a fall detection system that can run directly on the device. In this study, an initial-level deep learning model for fall detection is deployed along with subsequent models using ensemble learners, and it is trained on the publicly accessible MobiAct dataset. A comparative analysis is drawn between initial (Convolution Neural Networks) and final predictors (Ensemble Learners). The health and well-being of elderly people can be considerably improved by the ISBFD model, which makes it possible to detect falls and promptly warn carers with accuracy upto 93% approximately.

Keywords— Smart homes, Fall detection, CNN, Ensemble, Inertial sensors.

I. INTRODUCTION

The risk of falls among people 65 and older is a major public health problem. A fall refers to an unplanned and sudden drop from a position of elevation to a lower one. According to the World Health Organization's assessment, falls account for as much as 40% of injury-related fatalities in the elderly population globally [1]. Consequently, falls rank as the second leading cause of death following automobile accidents. Furthermore, falls are responsible for 60% of head injuries and 90% of hip and wrist fractures among the elderly population. Elderly people's mental health might suffer from frequent falls, which can undermine their self-confidence and ability to live independently [2]. Given the circumstances, it is imperative that older people remain in their homes for as long as they can, unless they have a condition that necessitates hospitalisation and endangers their lives. However, an accurate system must be in place to allow for remote health checks before this can be accomplished. Smart home healthcare has recently been a successful method for remotely monitoring elderly persons who live at home [3]. The well-being of elderly individuals is tracked by this technology using a range of devices, including as infrared sensors, mouthpieces, cameras, pressure sensors, and wearable sensors [4-6]. Then, in order to promote better illness management and ensure the elderly's well-being while living at home, this information is disseminated to medical professionals and carers. One of the most essential services provided by assisted home care is fall detection. However, commercial fall detection systems have a limited field of view and are expensive to install and operate. In addition, privacy issues are raised by video-based fall monitoring systems. To improve care for the elderly, it is crucial to create a fall detection system that is more affordable, automated, flexible, and dependable. The use of smartphones with sensors like accelerometers and gyroscopes by elderly persons is on the rise. Using the information gathered from various sensors, this technique allows for the detection of falls. Smartphones with sensors, as compared to wearable sensors, are adaptable and convenient. Elderly persons may not feel comfortable wearing sensors, but since they are more likely to carry smartphones, fall detection on smartphones is an easy alternative.

This paper suggests the use of the ISBFD framework which uses a platform for smart home healthcare that is assisted to identify falls in real time and enable help requests. The framework makes use of accelerometer sensor data from a smartphone, which is then processed and examined by a fall detection system that can runs online on the phone. Smartphone-based fall recognition systems have been thoroughly investigated by researchers in earlier works [7-9]. In these research, falls were detected using features gathered from smartphone sensors using threshold-based decision algorithms [10]. Although developing threshold-based decision algorithms is easier and requires less processing, selecting the appropriate threshold values requires striking a careful balance between identifying all falls (true positives) and avoiding classifying routine actions as falls (false positives). Finding thresholds that reliably function for everyone is challenging. Support vector machines [7] and Artificial neural networks [8] are two machine learning techniques that have recently been utilised to categorise falls from daily activities. Cahoolessur et al. [11] created a machine learning model based on waist-worn wearable devices using the XGBoost algorithm and the Sisfall dataset. To forecast falls, the model makes use of acceleration data that has been preprocessed into features.

Although these methods have generated good results, deeper understanding can be attained by applying deep learning models that make use of the substantial quantity of data that can be gathered from mobile sensors. In the fall detection system, Wang et al. [12] proposed a multi-source CNN integrated structure where the data from the pressure sensor, acceleration sensor, and gyroscope are individually preprocessed and formatted. Another study in [13] presented experimental model to enhance the care provided to elderly residents using a context-aware sensor system (CARE) for nurses in nursing homes through an Android tablet application. Long short-term memory (LSTM) network [15] and a Convolutional neural network (CNN) [14] have been combined to create a deep CNN ensemble model, which is an ensemble based learning model that has been created in the proposed ISBFD framework. The accelerometer sensor in smartphones is used to derive localised features. A dropout method is used to arbitrarily neglect a few neurons during training in order to prevent overfitting. The suggested model is trained using offline data, and it can then be integrated with smartphones for online, real-time fall detection. Using accelerometer data gathered from the publicly available dataset MobiAct, the ISBFD framework is validated for real-time fall detection use cases [16]. According to experimental findings, the suggested framework can categorise falls and non-falls more precisely. The rest of this article is organized as follows. First, we will discuss the proposed framework for detecting falls in the elderly. Then, we go over each component of this structure in depth. Following that, we brief the proposed approach for real-time fall detection, which is based on deep learning and ensemble models. Finally, we give the experimental findings that indicate the effectiveness of the suggested deep CNN-based ensemble technique.

II. METHODOLOGY

The primary components of the ISBFD architecture are depicted in Fig. 1. This system operates by collecting live data from a smartphone's accelerometer, gyroscope, and orientation sensors, which are then analyzed and processed by an integrated real-time fall detection system. Subsequently, the smartphone can generate an auditory alert via a Wireless Application Protocol (WAP) to notify family members and an SMS alert to inform caregivers via a mobile network.

A. Data processing and Feature extraction module

In this module, a 200-record window size is used to analyze and normalize the signal of human activity detected by smartphone sensors in real-time. This window size is sufficient to analyze each activity. With respect to the accelerometer sensor, 15 features are calculated from each of its three axis, for a total of 45 features. Fig. 2 illustrates and lists all of the attributes computed, including maximum value, maximum absolute value, minimum value, and minimum absolute value.

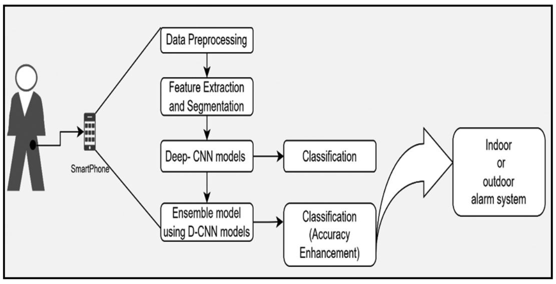

Fig. 1. ISBFD framework

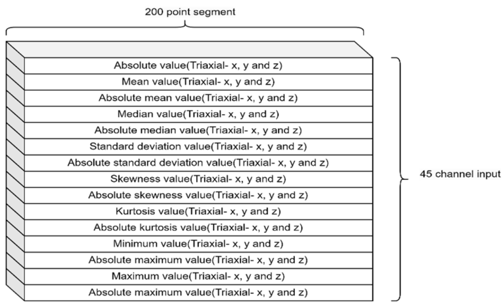

Fig. 2. 45-channel input feature set

B. CNN Architecture

In our model, we followed a typical Convolutional Neural Network (CNN) architecture. CNNs are basically group of three main components or layers: convolutional, pooling, and fully-connected layers. For experiments we fragmented data into segments having size 200 X 45, which contain 200 record points and 45 features. Convolution operations are used to create the convolution layers of CNN models, which are then followed by the max-pooling layer. Next layer of CNN is activation function, it takes inputs from previous convolution layer. Rectified Linear Unit (ReLU) activation function is used in this study. The last layer of the CNN is a Fully-Connected (FC) layer. The output from FC layer is then fed into a softmax function, which produces a categorization over the 2 classes, falling and non-falling activities of daily life. Our model's performance was evaluated using a dataset of 4000 samples.

The study evaluates four different CNN models for fall recognition. Model I has 1 convolutional layer and one fully-connected layer. Model II has 4 convolutional layers and 1 fully-connected layer, using 8 x 1 convolutional and pooling kernels. Model III is a wide neural network consisting of 5 convolutional layers and 1 FC layer. The first convolutional layer has 96 kernels having size as 8 and stride as 1 on a 200 x 45 input segment. The second layer has 256 kernels, and the third and fourth have 384 kernels each. The fifth layer has 256 kernels. A max-pooling layer of size 8 and a ReLU activation function are present in every convolutional layer. Both max-pooling and convolution kernels use the same amount of padding. This prevents differences in input and output size [17]. Model IV has six convolutional layers with 1 fully-connected layer and kernels of sizes 128, 256, 512, 1024, and 512. In all models, dropout is used to prevent overfitting.

The CNN models used in the study are thoroughly described in Table I.

Table I. CNN model architecture

Model name	1st Layer	2nd Layer	3rd Layer	4th Layer	5th Layer	6th Layer		Total Layers
Model I	4 kernels- conv 8	-	-	-	-	-	Dropout and Fully Connected Layer	**1+1=2**
	Max-pooling size=4	Max- pooling size=4	-	-	-	-		
Model II	4 kernels- conv 8	8 kernels- conv 8	16 kernels- conv 8	32 kernels- conv 8	-	-		**4+1=5**
	Max- pooling size=8	Max- pooling size=8	Max- pooling size=8	Max- pooling size=8	-	-		
Model III	96 kernels- conv 8	256 kernels- conv 8	384 kernels- conv 8	384 kernels- conv 8	256 kernels- conv 8	-		**5+1=6**
	Max- pooling size=8	Max- pooling size=8	Max- pooling size=8	Max- pooling size=8	Max- pooling size=8	-		
Model IV	128 kernels- conv 8	256 kernels- conv 8	512 kernels- conv 8	1024 kernels- conv 8	1024 kernels- conv 8	512 kernels- conv 8		**6+1=7**
	Max- pooling size=8	Max- pooling size=8	Max- pooling size=8	Max- pooling size=8	Max- pooling size=8	Max- pooling size=8		

C. Ensemble Models

The process of combining predictions from multiple machine learning models is known as an ensemble classifier. Such classifiers have potential of providing high accuracy results when their performance is compared against single classifiers. Some popular ensemble methods such as boosting, bagging and stacking are widely used. We have used stacking as one of our ensemble where we combine the outputs of a set of base learners and using another algorithm, known as the meta-learner, to make the final predictions. This approach can help improve the accuracy of the final predictions by combining the strengths of multiple models [18]. Ensemble methods are widely used in neural networks to improve the accuracy of predictions. Two commonly used ensemble methods are average voting and majority voting. The approach is to calculate the average or majority of the predicted probabilities from every model is used to generate a posterior label. Alternatively, the most common predicted label from all the models can be selected as the final prediction. Another method involves assigning weights to each model based on their performance, and combining their predictions using these optimal weights. After that using the weighted prediction from each model final prediction is made.

In summary, the study explores four different methods for combining the outcomes of multiple CNN models: majority, average voting, optimal weights, and logistic regression as meta-learner. The optimal weights are calculated by reducing the mean squared error function of the output coming from the base learners, and best-performing classifiers are assigned comparatively high weights. In order to improve the accuracy of CNN models further, ensembles are used. Here the CNN models are referred to as base learners (initial predictors) and the models used to make the resultant predictions are referred to as subsequent models.

III. EXPERIMENTS AND DISCUSSIONS

The proposed framework was evaluated on a laptop running on a Windows operating system, equipped with an i7-4510U CPU (2.0 GHz) processor and 8 GB RAM. The implementation was carried out using the Python programming language on Google Colab. The performance of the model was evaluated using the accuracy measure.

$$\text{Accuracy} = (TP + TN)/(TP + FP + TN + FN) \quad (1)$$

In equation (1), the accuracy measure is defined as the ratio of correctly recognized samples to all samples in the testing set. The measure includes true positive (TP) and false positive (FP) rates, as well as true negative (TN) and false negative (FN) rates. The next subsections provide information on the materials used in the experiments, such as the dataset, and the results are presented with comparisons and discussions.

A. Data Description

In our experiment, we utilized the MobiAct dataset to evaluate the proposed ISBFD framework. Biomedical Informatics and Health lab collected and published data publically for non-commercial research and educational purposes [16]. The raw data used in the study was collected from Samsung Galaxy S3 smartphones placed in the side pockets of participants. The data was gathered from accelerometers, gyroscopes, and orientation sensors. MobiAct dataset contains recordings from 67 subjects performing 4 types of falls, 11 types of daily living activities, and a lying activity after a fall. Both male and female subjects were included in the dataset, ranging in age from 20 to 47 years, height from 160 to 189 cm. For the case study, a portion of the raw accelerometer signal data was used, which included 4 different falls and 2 activities of daily life (viz. standing and lying) during the inactive period after a fall activity has been observed. Table III in the study provides more detailed information on the activities of daily life and falling acts included in the dataset.

B. Hyperparameters Used

In the initial experiment, we tested different types of CNN structures to see which ones works better for our task. We tried various combinations of layers, kernels, pooling size, optimization methods, stride size, batch sizes and activation functions. We found that for our problem Adam optimize works well when experimented with other optimizers such as gradient descent. Learning rate is set to 1e-4 for all the CNN models. We experimented with 50, 100, and 200 epochs, and we applied early stopping techniques and dropout technique to prevent model from overfitting. We provided varied batch sizes of like 16, 32, 64, and 128, and we concluded that a batch size of 64 provided better results. Finally, we compared three activation functions: 'tanh', 'Relu', and 'Leaky Relu', and we found that 'Relu' gave the best results. The initial predictors in this study consist of models I to IV (refer to Table I), which were developed using various CNN architecture types and their corresponding values used as hyper- parameter are indicated in Table II. Afterward, each method in the subsequent predictors is used to predict the final labels based on the output generated by the initial predictor models.

Table II. Hyperparameter values used

Hyper-Parameter	Value used	
Max- Pooling layer	Pooling size: 2, 4, 8	
Batch size	64	
Epoch Iterations	50,100,200	
Activation Layer-Output	Softmax	
Optimizer	Adam	
Conv Layer	Stride	1,2
	Activation function	ReLU
	Kernel size	2, 4, 8
	# kernels	4,8,16,32,96,128,256,384,512,1024

Table III. Activity Description of Mobiact Dataset

S.No	Activity	Description	Trails	Time Duration(sec)
1	Forward-lying	A forward fall from a standing position where the individual uses their hands to reduce the impact of the fall.	3	10
2	Front-knees-lying	Upon falling forward from a standing position, the initial impact is observed on the knees.	3	10
3	Back-sitting chair	A fall occurring during an attempt to sit on a chair, with a backward motion.	3	10
4	Sideward-lying	A sideways fall from standing position while bending legs.	3	10
5	Standing	Stationary stance with minor perturbations	1	300
6	Walking	Normal walking.	1	300
7	Jogging	Jogging.	3	30
8	Jumping	Continuous jumping.	3	30
9	Stairs up	Ascending a 10-step staircase.	6	10
10	Stairs down	Descending a 10-step staircase.	6	10
11	Stand to sit (sit on chair)	Transition from standing to sitting	6	6
12	Sitting on chair	Sitting on a chair with minor perturbations	1	60
13	Sit to stand (up from chair)	Transition from sitting to standing.	6	6
14	Car-step in	Stepping-in inside car.	6	6
15	Car-step out	Stepping-out of car.	6	6
16	Lying	An activity performed during the period of lying down following a fall.	12	NA

Fig. 3 shows the train and test accuracy as well as the loss function of Model IV. The train accuracy rises as the when we increase number of epochs, while the test accuracy reaches a stable point at around 100 epochs.

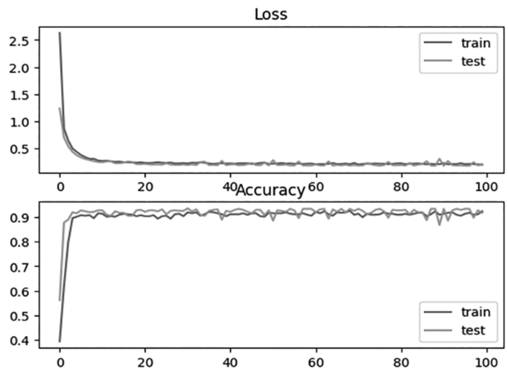

Fig. 3. Model IV Accuracy and Loss Curves (Train and Test)

C.Comparision with Ensemble Variants

Table IV illustrates the accuracy obtained by CNN models and their variations along with the performance of ensemble learners. With a prediction accuracy of 92.72%, the optimal weight technique outperforms even the best performing CNN model, Model IV. When Random Forest is used as meta-learner in stacking ensemble, highest prediction accuracy of approximately 93.51%, is obtained, indicating a better performance. Average voting ensemble did not perform well when compared with model IV but similar performance has been observed when compared with majority voting ensemble.

IV. CONCLUSION AND FUTURE WORK

It's critical to recognise falls accurately and immediately for ambient assisted smart living in healthcare, especially for elderly adults. This article introduces a fall recognition framework enabled by smartphones that identifies falls by continuously and automatically analyzing data from the smartphone's inertial sensors. In order to detect falls in real-time, we used a deep ensemble learning model that was trained offline on the publicly available MobiAct dataset. The proposed system can predict the activity into two categories namely non-falling act and falling act with more precision.

The ability of various convolutional neural networks (CNNs) to predict falls in elderly people has been subjected to the test. The raw sensor data was divided into 45 segments, each of which was of 200 points in size after pre-processing. To achieve high prediction accuracy, we studied different CNN architectures, combined the results using ensemble methods, and then looked into further CNN architectures. We used a number of CNN models in our ensemble model, each with different hyper-parameter values and configurations. Using "majority voting," "average voting," and each classifier's optimal weight, the outcomes of CNN models were combined. As a meta-learner, we also used Logistic Regression (LR). Different ensemble strategies that we tested all outperformed standalone CNN models. The ensemble methodology using LR as a meta-learner outperformed the other three methods as well, with an accuracy of 93.51%. Both the majority vote and the optimal weight techniques predicted percentages that were roughly in the range of 91-92%. In the future, we aim to integrate more sensor data, such as GPS data to track users' movements and integrate monitoring of other health related activities.

Table IV. Test accuracy of all experimented models

Model	Model I	Model II	Model III	Model IV	Ensemble Model (Average Voting)	Ensemble Model (Majority Voting)	Ensemble Model (Optimal weight)	Ensemble Model (LR as meta-learner)
Accuracy(%)	73.29	78.53	86.16	91.83	90.74	91.46	92.72	93.51

REFERENCES

[1] Todd, C., & Skelton, D. (2004). What are the main risk factors for falls amongst older people and what are the most effective interventions to prevent these falls?. World Health Organization. Regional Office for Europe.

[2] Fortino, G., & Gravina, R. (2015). Fall-MobileGuard: A smart real-time fall detection system. In Proceedings of the 10th EAI International Conference on Body Area Networks (pp. 44-50).

[3] Kvedar, J., Coye, M. J., & Everett, W. (2014). Connected health: a review of technologies and strategies to improve patient care with telemedicine and telehealth. Health affairs, 33(2), 194-199.

[4] Wan, J., Zou, C., Ullah, S., Lai, C. F., Zhou, M., & Wang, X. (2013). Cloud-enabled wireless body area networks for pervasive healthcare. IEEE Network, 27(5), 56-61.

[5] Hossain, M. S., & Muhammad, G. (2019). Emotion recognition using deep learning approach from audio–visual emotional big data. Information Fusion, 49, 69-78.

[6] Hossain, M. S., Muhammad, G., & Alamri, A. (2019). Smart healthcare monitoring: a voice pathology detection paradigm for smart cities. Multimedia Systems, 25, 565-575.

[7] Albert, M. V., Kording, K., Herrmann, M., & Jayaraman, A. (2012). Fall classification by machine learning using mobile phones. PloS one, 7(5), e36556.

[8] Kerdegari, H., Mokaram, S., Samsudin, K., & Ramli, A. R. (2015). A pervasive neural network based fall detection system on smart phone. Journal of Ambient Intelligence and Smart Environments, 7(2), 221-230.

[9] Hassan, M. M., Uddin, M. Z., Mohamed, A., & Almogren, A. (2018). A robust human activity recognition system using smart-

phone sensors and deep learning. Future Generation Computer Systems, 81, 307-313.

[10] Yu, D., & Li, J. (2017). Recent progresses in deep learning based acoustic models. IEEE/CAA Journal of automatica sinica, 4(3), 396-409.

[11] Cahoolessur, D. K., & Rajkumarsingh, B. (2020). Fall detection system using XGBoost and IoT. R&D Journal, 36, 8-18.

[12] Wang, L., Peng, M., & Zhou, Q. (2020). Pre-impact fall detection based on multi-source CNN ensemble. IEEE Sensors Journal, 20(10), 5442-5451.

[13] Klakegg, S., Opoku Asare, K., van Berkel, N., Visuri, A., Ferreira, E., Hosio, S., ... & Ferreira, D. (2021). CARE: Context-awareness for elderly care. Health and Technology, 11, 211-226.

[14] Hsieh, Y. Z., & Jeng, Y. L. (2017). Development of home intelligent fall detection IoT system based on feedback optical flow convolutional neural network. Ieee Access, 6, 6048-6057.

[15] Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2016). LSTM: A search space odyssey. IEEE transactions on neural networks and learning systems, 28(10), 2222-2232.

[16] Vavoulas, G., Chatzaki, C., Malliotakis, T., Pediaditis, M., & Tsiknakis, M. (2016, April). The MobiAct dataset: Recognition of activities of daily living using smartphones. In ICT4Ageing-Well (pp. 143-151).

[17] Dumoulin, V., & Visin, F. (2016). A guide to convolution arithmetic for deep learning. arXiv preprint arXiv:1603.07285.

[18] Ju, C., Bibaut, A., & van der Laan, M. (2018). The relative performance of ensemble methods with deep convolutional neural networks for image classification. Journal of Applied Statistics, 45(15), 2800-2818.

[19] Caruana, R., Lawrence, S., & Giles, C. (2000). Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. Advances in neural information processing systems,