Journal of Information Technology Management

Print ISSN: 2008-5893

Online ISSN: 2423-5059

A Multimodal Approach of Machine and Deep Learnings to Enhance the Fall of Elderly People

Saleh Al meraikhi*

*Corresponding author, College of Engineering, Abu Dhabi University, UAE. Email: 1078150@students.adu.ac.ae

Murab Al - Rajab

College of Engineering, Abu Dhabi University, UAE. E-mail: murad.al-rajab@adu.ac.ae

Abstract

Falls are a serious concern among the elderly due to being a major cause of harm to their physical and mental health. Despite their potential for harm, they can be prevented with proper care and monitoring. As such, the motivation for this research is to implement an algorithmic solution to the problem of falls that leverages the benefits of Machine Learning to detect falls in the elderly. There are various studies on fall detection that works on one dataset: wearable, environmental, or vision. Such an approach is biased against low fall detection and has a high false alarm rate. According to the literature, using two datasets can result in high accuracy and lower false alarms. The purpose of this study is to contribute to the field of Machine Learning and Fall Detection by investigating the optimal ways to apply common machine and deep learning algorithms trained on multimodal fall data. In addition, it has proposed a multimodal approach by training two separate classifiers using both Machine and Deep Learning and combining them into an overall system using sensor fusion in the form of a majority voting approach. Each trained model outputs an array comprising three percentage numbers, the average of the numbers in the same class from both arrays is then computed, and the highest percentage is the classification result. The working system achieved results were 97% accurate, with the highest being achieved by the Convolutional Neural Network algorithm. These results were higher than other state-of-the-art research conducted in the field.

Keywords: Machine Learning; Deep Learning; Fall Detection; Elderly People; Multimodal; Sensors; Video; Healthcare.

Journal of Information Technology Management, 2022, Vol. 14, No.3, pp. 168-184
Published by University of Tehran, Faculty of Management
doi: https://doi.org/ 10.22059/jitm.2022.88290
Article Type: Research Paper

Received: October 05, 2021
Received in revised form: February 26, 2022
Accepted: March 21, 2022
Published online: July 13, 2022

Introduction

Falls are a serious concern among elderly people, as their physical conditions can often not handle injuries caused by them. Approximately 684,000 people die from falls every year and most of the victims are elderly people aged 60 and up(WHO, 2021). As medical technologies advance every year, more and more elderly people will be able to live longer lives, but this comes at the cost of weakening physical conditions. It is expected that the global elderly population will rise to a total of 1.5 billion by 2050(UN, 2019). This rise in numbers creates more opportunities for falls to occur, leading to psychological, physical, and financial difficulties for the victim and their families. In addition, treating fall victims puts a strain on medical infrastructure and resources. With, there needs to be new research conducted to find ways to mitigate this problem. One area that shows promise is using Machine Learning to perform fall detection. Machine Learning is a rapidly growing field, and it has many applications in various fields such as finance, technology, and medicine. It is estimated that Artificial Intelligence and Machine Learning market in the healthcare field will rise to 28 billion U.S. dollars by the year 2025(C.Stewart,2020). Given the same input data, Machine Learning systems are often able to detect falls with greater accuracy and efficiency than humans, when it comes to Fall Detection. This is an emerging field of research in the Machine Learning domain and shows potential for revolutionizing elderly care. There are multiple types of fall detection systems, as follows: Wearable systems, where falls are detected via sensors attached to the patient's body, have been shown to achieve high results(Wu et al., 2019)(Nooruddin et al., 2020). Environmental systems use sensors placed in the patient's surroundings to detect the fall, by measuring spikes in different sensor types such as audio, infrared, or light intensity (Z. Chen & Wang, 2018),(Srp & Vajda, 2012) and (Yun Li et al.,2012). Vision-based fall detection uses cameras to capture video frames of the person and analyze them for falling action (Lezzar et al., 2020) and (Ariunbold et al., n.d.). However, a technique to combine all three of the other types exists in the form of multimodal fall detection, It is a combination of elements from wearable, environmental, or vision-based systems. Combining audio and acceleration sensor readings(Geertsema et al., 2019) or sensor fusion(Haobo Li et al., 2020),(Xu et al., 2021) is an example. To implement a solution to this problem, this research expands upon the fall detection system which we have implemented, which is a multimodal fall detection system. Our approach trains two different models, one for the sensor readings of the fall, and one for the video frames of the fall. Both models are combined into one system by averaging the results of both systems to generate a result. This method of implementation was selected because it is a simple and flexible way to combine two different classifiers for fall detection and has been used in other research achieving good results (A.diete et al., 2019),(Sexena et al.,2021). In addition, this approach has the benefit of being able to account for noise or errors since any abnormal readings in one source can be compensated by the other.

This paper is organized as follows: Section 2 will expand more upon the concept of fall detection and summarize existing research in the field. Section 3 will discuss the materials and methods for the research, while section 4 summarizes the results and discuss the findings. Finally, the paper will conclude with the conclusion and future work in section 5.

Literature Review

Tri-axial accelerometers are a common device used in wearable fall detection systems used to collect input data. By measuring the changes of the acceleration components in 3 dimensions, the quantity spikes caused by falls can be detected(Wu et al., 2019),(Li et al., 2018). The accelerometers may be an Inertial Measurement Device (IMU)(Nooruddin et al., 2020) or the built-in sensors found in modern smartphones(Zia et al., 2020). The measurements can be fed to a classifier trained using common algorithms such as Convolutional Neural Networks or Decision Trees(Yacchiremma et al.,2018). Another approach is to use a variety of sensors for data capture and create a covariance matrix for analysis in a technique known as sensor fusion(Boutellaa et al., 2019).

Environmental approaches to fall detection involve the placement of sensors in the surrounding area of the patient to capture input signals. The signals may be of the ambient sound(Adnan et al., 2018), ambient light(AM srp et al.,2012) or infra-red(Z. Chen & Wang, 2018). The main advantage of using them is that it lacks the discomfort caused by body sensors but suffer from the problems of environmental noise.

Using a vision-based approach to collect the video frames of a falling person, multiple fall detection techniques are possible. Deep Learning approaches using Convolutional Neural Networks (CNNs) have been shown to achieve good results(Ariunbold et al., n.d.),(S. S P & A.J, 2019). CNNs also help with the complexities of integrating multiple cameras to the vision-based approach by achieving high results, as shown in the work of (Espinosa et al., 2020), whose approach using RF, KNN, SVM, and MLP algorithms achieved 95.09%. Depth image extraction creates a depth map of the input frames to detect areas of sensitivity when a fall happens(Panahi & Ghods, 2018). Filter based approach utilizes common filtering algorithms such as the Kalman or Gabor Filters to measure angles between the ground and a falling person. The work done in (Sangeetha et al., 2020) is an example of this method, where ground point estimation using the Gabor Filter and movement tracking using the Kalman filter using the University of Rzeszow (UR) dataset generated 90% accuracy. It is possible to implement vision-based systems using models already trained on other computer vision tasks instead of training them from scratch. In this technique, known as transfer learning, the pre-trained models' weights are re-tuned and re-configured for a fall detection setting. This approach saves time on the training step and has been shown to achieve good results when applied to this problem domain, as shown by (Sciences, 2021) achieving 98% using a deep learning approach using the AlexNet neural network and (N.Nahar et al., 2020) 97% accuracy using C4.5 classifier.

Multimodal systems show considerable differences in the types of modes applied. While most of them use 2 modes, some implement more complex systems involving multiple modes. An example is a work of (Martinez-Villaseñor & Ponce, 2020), who used a total of 7 different modes and a variety of machine learning algorithms that achieved 95% accuracy using the Random Forest model. Multimodal systems introduce the added complexity of working with the different types of input data, and as such require more complex applications of algorithms to analyze the inputs for fall detection. Deep Learning models have been used

for this solution, such as the work of (de Assis Neto et al., 2020) which used Long Short-Term Memory Networks and got 70% accuracy using multimodal inputs. Another method is to use hybrid neural networks involving multiple input layers as implemented in (H Li et al., 2019). Their results were 97% accuracy using a hybrid fusion multimodal approach that utilized environmental and wearable sensors. Most voting approaches combine classifiers by averaging the results from multiple models, as shown by (Saxena et al., 2021). Their approach using ensemble learning which combines Support Vector Machine, K Nearest Neighbors, Decision Trees, and Random Forest achieved 87.42% accuracy. Another example is the sensor fusion technique performed by (Diete et al., 2019), which used both early and late Decision Level fusion from acceleration and video inputs, a Random Forest and Logistic Regression model were fused to achieve 79.6% accuracy.

Methodology

Explanation of Model

First, fall detection data will be taken from public datasets, which will be preprocessed. Preprocessing is done to transform the data to make it suitable for being used by Machine Learning algorithms for the training step as well as to format them to a common standard. Some processes involved in this step include feature Scaling and Augmentation. The latter scales the input data down into a smaller range (between 0 and 1) while the latter modifies each data point with slight variations to produce a more diverse set. Once the overall data from the dataset is prepared, it is split into a training and testing set. After that, it is ready to be fed into the machine learning algorithms for training. Since this is a multimodal classification system, two different classifiers are used: one model to train the images and the other for the sensor data. Figure 1 shows the framework for the proposed system, showing the process from start to end.

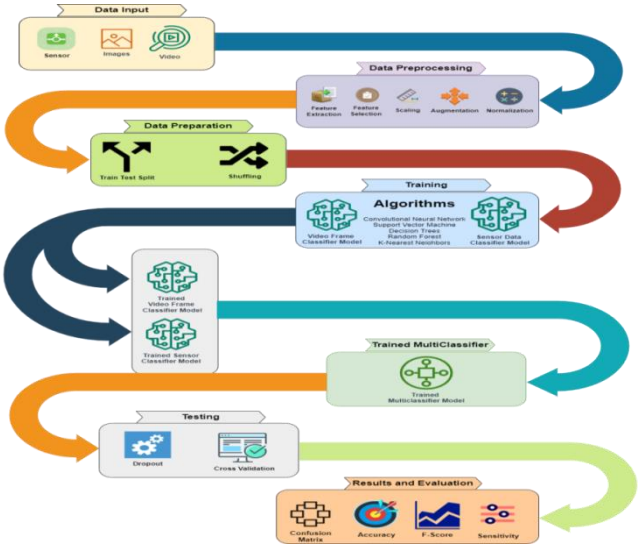

Fig.1. Proposed System Framework

The two trained models are combined into making a multimodal classification using the fusion technique of majority voting, which is working as follows:

- The video and sensor model will each return an array of confidence levels. The confidence level denotes the probability that the sample being classified by the model belongs to a class. In our model, the format of the array is [FALL, FALLING, NOT FALL], where FALL represents that a fall has taken place, FALLING represents the subject is in the process of falling down and NOT FALL denotes no fall has occurred. Each number will be a percentage value, and the highest number is the category the sample is likely to be classified. For example, if one model returns the array [10%, 90%, 1%], then we can say that the current sample belongs to the FALLING category since the highest percentage value is the second number in the array.
- We take the average of both arrays returned by each of the models by averaging the numbers of each position in the array. So, this means we average the first number in the first array with the first number of the second array to get the average confidence level of the FALL category and repeat it for the second and third positions. This will return a single averaged array of confidence levels of both models.
- The highest result in the array is taken as the classification result.

The following is an example calculation using this system:

- We have the video model array A1, with its three percentage values x1, x2, x3 as shown in

$$A1 = [x1, x2, x3] \quad (1)$$

- We have the video model array A2, with its three percentage values y1, y2, x3 as shown in

$$A2 = [y1, y2, y3] \quad (2)$$

- We average each number with the value in the corresponding position in the second array in Equation 3 to get the result A3:

$$A3 = A1 + A2 = [\ (y1 + x1)/2\ ,\ (y2 + x2)/2\ ,\ (y3 + x3)/2\ ] \quad (3)$$

- The resulting array after the calculation is A3 (Equation 4). It will contain 3 values z1, z2, z3 which represent the percentages for the categories FALL, FALLING, and NOT FALL, respectively:

$$A3 = [z1, z2, z3] \quad (4)$$

- The resulting class will be the maximum of z1, x2, z3 (Equation 5):

$$A3 = \max(z1, z2, z3) \quad (5)$$

In this method, averaging the results of each model allows us to get a better result for the classification by accounting for noise or any error sources. So, if one model is affected by any of these problems, by averaging the result, we can ensure that an accurate classification result is guaranteed. We use common classification algorithms such as Convolutional Neural Networks (CNN), Decision Trees (DT), and Support Vector Machines (SVM) among

others(X Wang et al., 2020). At this stage, the algorithms will use the prepared data to find patterns in the problem domain, which in this case is the detection of falls, to be able to make independent classifications on new data. Several parameters will be tuned here to achieve optimal results in the form of accuracy, sensitivity, and specificity among others. The training step will result in a trained multimodal classification model which is a combination of the two trained models. This model will be tested on new data so that it can make classifications for fall detection and prediction. The final step of the process will be to get the results of the testing step and evaluate the performance of the trained model using various metrics such as accuracy, and sensitivity among others.

The following steps take place in the procedure of the experiment conducted for the proposed model:

1. **Data Collection:** The input data is collected from the dataset. This involves downloading all the necessary files and importing them to be used in the program.
2. **Data Preprocessing:** Once the input data is captured it needs to undergo pre-processing before it can be fed into the classifier. Two important methods which occur here are feature selection and feature extraction. Feature selection is where only the necessary features are extracted from the input data. This is because there will be lots of different variables in the input data, and to avoid redundancy (where the same measurements are present for one quantity) and put less processing strain on the classifier it becomes necessary to only select the appropriate ones. On the other hand, feature extraction is the process where features are transformed or combined into new ones. For example, body sensors for fall detection will measure acceleration in three dimensions (x, y, and z planes) and feature extraction will combine the three measurements into one representing the net acceleration. An additional task is to clean the data of noise by removing data points that are abnormal or outside the acceptable range.
3. **Data Preparation:** Once processing is complete, the processed data is now prepared for training in the machine learning classifier. For this purpose, the data is split into testing and training sets. The purpose of the training set is to train the machine learning classifier on detecting falls. The ratio is 80 % for the training set and 20% for the testing set. The training set can be further split into a validation set, which is used during the training phase to ensure that the model does not overfit. Therefore, the resulting split will be 80-10-10 for the training set, testing set, and validation set respectively. Shuffling is another process that randomizes the orders of items in both sets to ensure that overfitting does not happen.
4. **Training:** The training process will apply common machine learning algorithms to train the classifier to detect falls. Two classifiers are trained and combined, one for training on sensor data and the other for training on video data
5. **Testing:** Once the training has been done, the testing set will be used to evaluate the performance of the classifier. The testing set contains the remaining data which was not a

part of the training set. Each trained model for each algorithm is tested on the test set, and the test accuracy of each one is generated.

6. **Evaluation:** The final step is to evaluate the performance of the overall system on how well it detects falls. Several metrics are used to judge its performance, such as accuracy or sensitivity. If the performance was not up to standard, then it becomes necessary to analyze and repeat the previous steps to find out the reasons for this and make adjustments. If the performance was acceptable, then the system can be deployed, or the previous steps can be repeated to give further improvements if required.

Dataset

The dataset used for this project comes from the University of Rzeszow (UR) Fall Detection Dataset, which was created by Bogdan Kwolek and Michal Kepski(Z. Chen & Wang, 2018). It is a multimodal dataset that is comprised of 70 videos of test subjects performing falls and daily activities. It is a common dataset in the research domain and is being used by many recent research papers such as (Al Nahian et al., 2021),(Sowmyayani et al., 2020),(Tahir et al., 2021), (Yhdhego et al.,2019) and (Nahar et al., 2020) which are recent. The data was collected by the researchers using 2 Microsoft Kinect cameras for recording videos and PS Move (60Hz) and x-IMU (256Hz) IMU devices for recording sensor data. The publicly available data consists of:

- 60 videos of falls conducted by test subjects from multiple camera angles.
- 40 videos of everyday activities conducted by test subjects.
- RGB frames of each video as PNG image files.
- Depth data for each image frame.
- Sensor data for each video frame along with their labels.

For this research, since the main focus was on Fall Detection, only the data related to falls was taken from the dataset. Specifically, the videos, the RGB frames, and sensor data; the data for the daily activity videos was ignored. For labels, the dataset classifies each frame into three categories:

- -1: Indicates the person in the frame is not lying on the ground.
- 0: Indicates the person in the frame is in the process of falling.
- 1: Indicates the person is lying on the ground.

The main reason for using the UR dataset is due to the extensive explanation of the data by the researchers, which makes it easy to work with, and since the data is neatly presented and formatted by default, requiring little configuration to bring into a Machine Learning setup for model training.

Parameter Settings

Table 1 shows the various parameters and settings used in the development of the system. The ones used here were selected after tuning each one to different settings until the optimal performance results were achieved. In addition, despite each algorithm having dozens of possible parameters available to tune, the ones selected here are the ones that have the most effect on results. There is a variation in the number of parameters for each setting. The K-Nearest Neighbors, for example, are only affected by their n-neighbors parameter while Convolutional Neural Networks have at least 7 which can have a significant impact.

Table 1. Summary of Experiment Parameters

Algorithm/Setting	Parameters
K-Nearest Neighbors (Sensor + Images)	n-neighbors=8
Support Vector Machine (Sensor + Images)	kernel=linear c-value=0.01
Decision Tree (Sensor + Images)	criterion=entropy max depth=3 random state=0
Random Forest (Sensor + Images)	n-estimators=100
Convolutional Neural Network (Image)	Optimization Function=Adamax Learning Rate=0.00001 Hidden Layer Activation Function=ReLu Output Layer Activation Function=SoftmaxEarly Stopping after 5 epochs Architecture=16 16 32 32 64 64 128 Dropout=30% in the Last Hidden Layer
Convolutional Neural Network (Sensor)	Optimization Function=Adamax Learning Rate=0.00001 Hidden Layer Activation Functions=ReLu Output Layer Activation Function=SoftmaxEarly Stopping after 5 epochs Architecture=8 16 32 64 128 Dropout=40%, 40%, 20%, 10%,10%
Train-Test Ratio (Sensors)	20% for testing, 80% for training
Train-Test Ratio (Images)	20% for testing, 10% for validation, 80% for training

Tools and Technologies

The tools and technologies used to implement the system are all open-source and available to use free of charge. They are all standard tools that are used in machine learning applications both in industry and academia. Each tool has a specific purpose, some are used for processing the data, some for applying the algorithms while others are used to import and manage the data. Most of the tools are based on the Python programming libraries which have extensive support for both Machine and Deep Learning applications.

- **TensorFlow**: TensorFlow contains software and functions for the design and implementation of Neural Networks for Deep Learning(Tensorflow, 2022).
- **Keras**: A Machine Learning API for Python through which the developer can interact with and implement Machine Learning and Deep Learning models(Keras, 2022).

- **Scikit Learn**: Python Libraries which contain implementations of common algorithms (K Nearest Neighbors, Support Vector Machines), data processing functions, and evaluation metrics(Scikit Learn, 2022).
- **OpenCV**: OpenCV is a computer vision library that comes with a set of functions for working with images and videos. It provides tools for editing and transforming them as well as computer vision functions for object detection, image recognition and other computer vision task(OpenCv, 2022).

Results

For the implementation, we conducted the experiments first with each classifier independently and recorded the results, and then we conducted the experiments using the proposed model. Table 2 shows the results of all experiments, while Figure 2 shows the comparison of the results in a bar chart.

Table 2. Accuracy Results of each Modal and for the Multimodal

	Sensor Accuracy	Video Accuracy	Sensor + Video Accuracy
Convolutional Neural Network	95%	99%	97%
Support Vector Machine	88%	93%	91%
K Nearest Neighbors	64%	77%	71%
Decision Tree	80%	85%	83%
Random Forest	89%	94%	91%

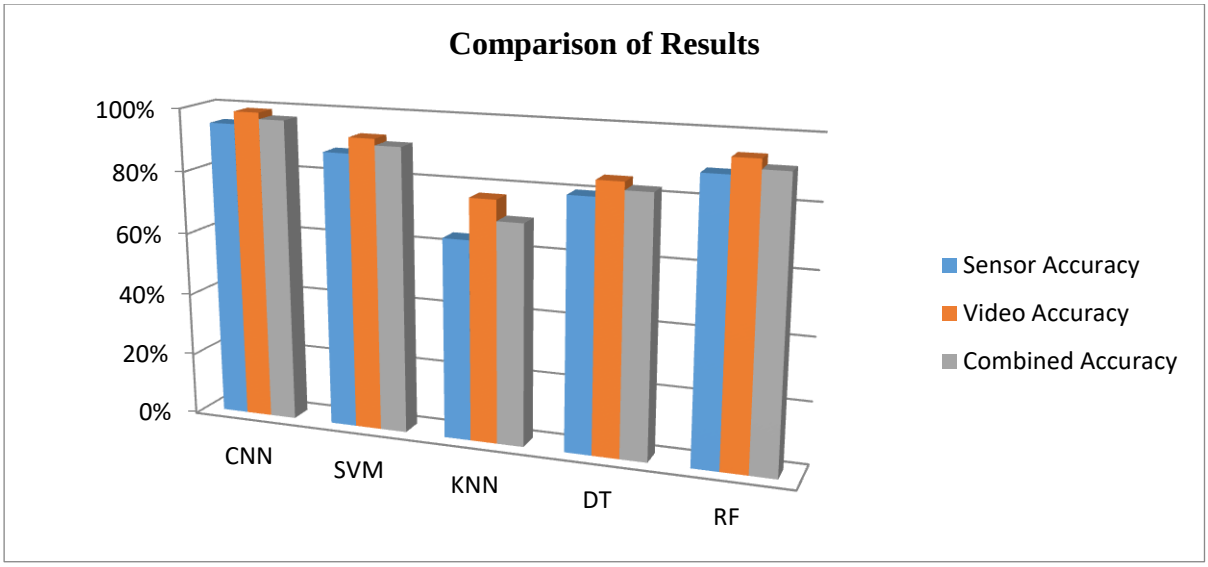

Fig.2. Comparison of Results(Bar Chart)

It is found that the highest accuracy was achieved by the CNN algorithm at 97%. All other algorithms except for KNN and DT achieved an accuracy of over 90 %. KNN achieved the worst performance is generally a less accurate algorithm than others and as such its results are expected. However, for decision trees and random forests, we see a big difference in the

accuracies. A Random Forest is composed of multiple Decision Trees, so it is expected that an RF will always perform significantly higher, however in our results, we see only a 4% improvement. This may be due to the low number of training samples or classes since both algorithms are not generally used for Classification tasks. We can see that there is a general trend of video accuracy being slightly higher than sensor accuracy, usually differing by 5-6%. However, this trend is not followed by the results from the KNN algorithm, which contains a 13% accuracy difference between its modalities. The sensor accuracy for KNN was 64%, which is the lowest out of all the results.

To measure the performance of our system, we used metrics of accuracy, recall, and precision. Before defining them, we must understand what our positive class and negative class are. In classification tasks, the positive class is the one we are aiming to detect (FALL in our case), while the negative class is the opposite of the positive class (NOT FALL in our case). Knowing this, we can first define our true and false categories for each class:

- TP: True Positive, which is when the positive class is correctly identified as the positive class.
- FP: False Positive, which is when a negative class is falsely classified as the positive class.
- TN: True Negative, which is the negative class correctly being identified as a negative class.
- FN: False Negative, which is when the positive class is falsely classified as the negative class.

Using these definitions, we can get the formulas to calculate the Recall, Precision, and Accuracy:

• **Recall**: Recall calculates the rate of True Positives to all positive classifications (i.e., TP and FN).

$$\frac{TP}{TP+FN} \tag{6}$$

• **Precision**: Precision refers to the accuracy of the positive class predictions. In other words, it measures the ratio of true positives to true and false positives.

$$\frac{TP}{TP+FP} \tag{7}$$

• **Accuracy**: It is showing the ratio of total correct classifications, for both the positive and negative classes, over the total classifications made.

$$\frac{TP+TN}{TP+TN+FP+FN} \tag{8}$$

Confusion Matrix

A confusion matrix shows the performance of the system in terms of the FN, TN, TP, and FP. It is a matrix of size n x n where n is the number of classes. In our case, since we have 3 classes, we will have a 3x3 matrix. The diagonal of the matrix represents the true classifications (i.e., True Positive and True Negatives) while the others represent the False Negative and False Positives. The rows represent the actual values, while the columns represent predicted values. The sum of values in each row must add up to the total number of items for that class. In our case, we have NOT FALLS, FALLING, and FALL categories.

Table 3 shows the confusion matrix for the K Nearest Neighbors Model. The NOT FALL class was identified correctly all the time since the FALLING and FALL values for that row are 0. The FALLING category was identified correctly 170 times and was incorrectly identified as NOT FALL 3 times, and as a FALL 7 times. The FALL category was identified correctly 180 times and was just identified as FALLING 1 time.

Table 3. K-Nearest Neighbors Confusion Matrix

	NOT FALL	FALLING	FALL
NOT FALL	238	0	0
FALLING	3	170	7
FALL	0	1	180

Table 4 shows the confusion matrix for the Random Forest Model. NOT FALL was only identified incorrectly once, as a FALLING class. FALLING was identified as NOT FALL 6 times and as a FALL 4 times. FALL was only identified incorrectly as a FALLING class one time.

Table 4. Random Forest Confusion Matrix

	NOT FALL	FALLING	FALL
NOT FALL	237	1	0
FALLING	6	170	4
FALL	0	1	180

Table 5 shows the confusion matrix for the Decision Tree Model. We can see that NOT FALL was identified as FALLING 13 times and as a FALL one time. FALLING was identified as a NOT FALL 51 times and as a FALL 59 times. FALL was identified as FALLING 3 times.

Table 5. Decision Trees Confusion Matrix

	NOT FALL	FALLING	FALL
NOT FALL	224	13	1
FALLING	51	70	59
FALL	0	3	178

Table 6 shows the confusion matrix for the Support Vector Machine Model. NOT FALL was classified incorrectly as a FALLING class twice. FALLING was incorrectly identified as a NOT FALL three times and as a FALL twice. FALL was identified as a FALLING class 6 times.

Table 6. Support Vector Machine Confusion Matrix

	NOT FALL	FALLING	FALL
NOT FALL	236	2	0
FALLING	3	175	2
FALL	0	6	175

Table 7 shows the confusion matrix for the CNN Model. NOT FALL was classified incorrectly as a FALLING class once. FALLING was incorrectly identified as a NOT FALL 1 time and as a FALL twice. FALL was identified as a FALLING class 2 times.

Table 7. Convolutional Neural Network Confusion Matrix

	NOT FALL	FALLING	FALL
NOT FALL	237	1	0
FALLING	1	177	2
FALL	0	2	179

From the confusion matrices, we can notice the following trends:

- FALLING class seems to be the one that is most wrongly identified in all algorithms. This is because the threshold between a FALLING/FALL or FALLING/NOT FALL can sometimes be very narrow. For example, when falling from a chair, the person will usually have his or her body moving closer to the floor, and the point at which the body is angled closer towards the ground can be arbitrary depending on the camera, lighting, and body position, etc.
- FALL class is the one that is least misidentified. This is a positive trend, as it means that we have less chance of missing a fall when it occurs.
- Convolutional Neural Networks, Random Forest, and K Nearest Neighbors models seem to be the ones that classify the correct class the most. This can be seen from the diagonal of their respective matrices, as the number is close to the total number of samples for each class. Support Vector Machine achieves medium performance, while Decision Trees achieve the worst performance since it has the greatest number of incorrect classifications for all classes.

Comparing Results with Previous Work

Table 8 shows the comparison of our table with other research papers which used the multimodal and multi-classifier approaches. Our accuracy uses the result achieved by the highest performing algorithm (CNN) at 97%.

- (A.diete et al., 2019) used two different methodologies an early fusion and late fusion

method. In late fusion, which is the one implemented in this research, the sensor and video classifiers are merged into a multi-classifier after each is trained separately. In early fusion, the multi-classifier is trained on both sensor and video data all at once. Using a neural network, both methods were tested, with the highest achieved at 79.6%

- (H Wei et al., 2020) achieved the highest accuracy of 95.1%. They performed segmentation on inertial sensor
- signals and video frames and trained each classifier separately using a 2D and 3D CNN respectively.
- The multi-classifier was implemented using an action score calculation which assigned weights to
- each modality.
- (H Wei et al., 2020) achieved the highest accuracy of 81.8%. They used an approach a 3D CNN is used for the
- video data and a 2D CNN is used for sensor data and combined into a fully connected
- classification layer using a sliding window technique.
- (H Wei et al., 2019) also used a 3D CNN for video and 2D for sensor data, achieving an accuracy of 95.6%.
- (M. Ehatisham et al., 2019) performed feature level fusion, where the sensor and RGB data were combined into one overall dataset using a feature vector and trained using the KNN and SVM algorithms. The highest performance was achieved by the KNN algorithm which got 89.3% accuracy

Table 8. Comparison of Results with previous Work in the Research Area

Paper	Accuracy
47	79.36%
50	95.1%
51	81.1%
52	95.6%
53	89.3%
Our Approach	**97%**

Conclusion

In this research paper, we analyzed the performance of common ML and DL algorithms when applied to the field of Multimodal Fall Detection. We found that a CNN Classifier which is composed of 2 separately trained networks, one for the sensor data and one for the image data can achieve relatively high performance (97%). It can be trained on images and sensor data for a simple fall detection setting to detect falls. To analyze the performance of the algorithms, we use the metrics of accuracy, precision and recall, with a greater emphasis

being placed on accuracy. We were able to increase the prediction accuracy of fall detection compared to similar research using the same dataset. The system can successfully combine two classifiers using the majority voting sensor fusion method to perform fall detection and prediction. In addition, the system can perform fall detection and prediction at the same time. In terms of our algorithm performance, our system is consistent with the majority of research in Fall Detection and Machine Learning, as the Deep Learning CNN approach outperforms traditional Machine Learning methods. Future work aims to optimize the algorithm training and testing procedure to increase accuracy. This could be done using better classifiers or by tuning the parameters to get optimal results. Several techniques already exist in the field of ML, such as cross-validation which runs the training multiple times with different hyperparameters to achieve the best result. Other methods include the use of object detection or pose estimation algorithms to detect falls by analyzing the orientation of the person's body of the use of transfer learning, which brings in a trained model originally applied to another problem domain and tunes the parameters to apply it to another one.

Conflict of interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article

References

Adnan, S. M., Irtaza, A., Aziz, S., Ullah, M. O., Javed, A., & Mahmood, M. T. (2018). Fall detection through acoustic Local Ternary Patterns. *Applied Acoustics*, *140*(March), 296–300. https://doi.org/10.1016/j.apacoust.2018.06.013

Al Nahian, M. J., Ghosh, T., Al Banna, M. H., Aseeri, M. A., Uddin, M. N., Ahmed, M. R., ... & Kaiser, M. S. (2021). Towards an accelerometer-based elderly fall detection system using cross-disciplinary time series features. *IEEE Access*, *9*, 39413-39431

Ariunbold, Y., Brito, S., & Leong, A. (n.d.). *FallDetectNet: A Computer Vision Platform for Fall Detection*.

Ariunbold, Y., Brito, S., & Leong, A. (n.d.). *FallDetectNet: A Computer Vision Platform for Fall Detection*.

based fall detection. *Journal of Intelligent Material Systems and Structures*, *29*(9), 2027–

based fall detection. *Journal of Intelligent Material Systems and Structures*, *29*(9), 2027–

Boutellaa, E., Kerdjidj, O., & Ghanem, K. (2019). Covariance matrix based fall detection from multiple wearable sensors. *Journal of Biomedical Informatics*, *94*(December 2018), 103189. https://doi.org/10.1016/j.jbi.2019.103189

C.Stewart(2019). Global market size for artificial intelligence in healthcare in 2016, 2017 and a

Chen, Z., & Wang, Y. (2018). Infrared–ultrasonic sensor fusion for support vector machine–

Chen, Z., & Wang, Y. (2018). Infrared–ultrasonic sensor fusion for support vector machine–

de Assis Neto, S. R., Santos, G. L., da Silva Rocha, E., Bendechache, M., Rosati, P., Lynn, T., & Takako Endo, P. (2020). Detecting Human Activities Based on a Multimodal Sensor Data Set Using a Bidirectional Long Short-Term Memory Model: A Case Study. *Studies in Systems, Decision and Control*, *273*, 31–51. https://doi.org/10.1007/978-3-030-38748-8_2

Diete, A., & Stuckenschmidt, H. (2019). Fusing object information and inertial data for activity recognition. *Sensors*, *19*(19), 4119.

Espinosa, R., Ponce, H., Gutiérrez, S., Martínez-Villaseñor, L., Brieva, J., & Moya-Albor, E. (2020). Application of Convolutional Neural Networks for Fall Detection Using Multiple Cameras. *Studies in Systems, Decision and Control*, *273*, 97–120. https://doi.org/10.1007/978-3-030-38748-8_5

forecast for 2025. Statistical, 22 October 2020

fusion,"IEEE Access, vol. 7, pp. 60736–60751.

Geertsema, E. E., Visser, G. H., Viergever, M. A., & Kalitzin, S. N. (2019). Automated remote fall detection using impact features from video and audio. *Journal of Biomechanics*, *88*, 25–32. https://doi.org/10.1016/j.jbiomech.2019.03.007

group of pictures," Vietnam Journal of Computer Science, vol. 08, no. 02, pp. 199–214

H. Li, A. Shrestha, H. Heidari, J. Le K., & F. Fioranelli(2020). Bi-LSTM Network for Multimodal Continuous Human Activity Recognition and Fall Detection, *IEEE Sensors Journal*, vol. 20, no. 3, pp. 1191-1201, doi: 10.1109/JSEN.2019.2946095.

https://www.statista.com/statistics/826993/health-ai-market-value-worldwide/

Journal of Engineering Research amp; Technology(IJERT), vol.8, no.6.

Keras(2022). https://keras.io/

Lezzar, F., Benmerzoug, D., Kitouni, I., Mehri, A., & Mendjeli, A. (2020). *Camera-Based Fall Detection System for the Elderly With Occlusion Recognition*. *42*(3), 169–179.

Li, H., Shrestha, A., Heidari, H., Le Kernec, J., & Fioranelli, F. (2019). Bi-LSTM network for multimodal continuous human activity recognition and fall detection. *IEEE Sensors Journal*, *20*(3), 1191-1201

Li, X., Nie, L., Xu, H., & Wang, X. (2018). Collaborative Fall Detection Using Smart Phone and Kinect. *Mobile Networks and Applications*, *23*(4), 775–788. https://doi.org/10.1007/s11036-018-0998-y

M. Ehatisham-Ul-Haq, A., Javed, M. A. Azam, H. M. Malik, A. Irtaza, I. H. Lee,& M. T.,

Mahmood(2019). Robust human activity recognition using multimodal feature-level

Martinez-Villaseñor, L., & Ponce, H. (2020). Design and analysis for fall detection system simplification. *Journal of Visualized Experiments*, *2020*(158), 1–11. https://doi.org/10.3791/60361

Nahar, N., Hossain, M. S., & Andersson, K. (2020, September). A machine learning based fall detection for elderly people with neurodegenerative disorders. In *International Conference on Brain Informatics* (pp. 194-203). Springer, Cham.

Nahar, N., Hossain, M. S., & Andersson, K. (2020, September). A machine learning based fall detection for elderly people with neurodegenerative disorders. In International Conference on Brain Informatics (pp. 194-203). Springer, Cham.

Nooruddin, S., Milon Islam, M., & Sharna, F. A. (2020). An IoT based device-type invariant fall detection system. *Internet of Things (Netherlands)*, *9*, 100130. https://doi.org/10.1016/j.iot.2019.100130

Nooruddin, S., Milon Islam, M., & Sharna, F. A. (2020). An IoT based device-type invariant fall detection system. *Internet of Things (Netherlands)*, *9*, 100130. https://doi.org/10.1016/j.iot.2019.100130

OpenCV(2022), 03-Feb-2022. https://opencv.org/

Panahi, L., & Ghods, V. (2018). Human fall detection using machine vision techniques on RGB–D images. *Biomedical Signal Processing and Control*, *44*, 146–153. https://doi.org/10.1016/j.bspc.2018.04.014

S. S P & A.J.(2019). Human Fall Detection using Convolutional Neural Network, International

Sangeetha, D. P., Vijayalakshmi, S., Arunachalam, S., Kokila, K., Prakash, U., & Swetha, S. (2020). Fall detection for elderly people using video-based analysis. *Journal of Advanced Research in Dynamical and Control Systems*, *12*(7 Special Issue), 232–239. https://doi.org/10.5373/JARDCS/V12SP7/20202102

Saxena, U., Moulik, S., Nayak, S. R., Hanne, T., & Sinha Roy, D. (2021). Ensemble-based machine learning for predicting sudden human fall using health data. *Mathematical Problems in Engineering*, *2021*.

Sciences, S. (2021). *A Framework For Fall Activity Detection and Classification using Deep Learning Method*. *1*, 56–65.

Scikit-Learn(2022). https://scikit-learn.org/

Sowmyayani, V. Murugan, & J., Kavitha(2020), Fall detection in elderly care system based on a

Srp, Á. M., & Vajda, F. (2012). Fall detection for independently living older people utilizing machine learning. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, *45*(18), 79–84. https://doi.org/10.3182/20120829-3-HU-2029.00011

Srp, Á. M., & Vajda, F. (2012). Fall detection for independently living older people utilizing machine learning. *IFAC Proceedings Volumes*, *45*(18), 79-84.

Tahir, A., Ahmad, J., Morison, G., Larijani, H., Gibson, R. M., & Skelton, D. A. (2021). Hrnn4f: Hybrid deep random neural network for multi-channel fall activity detection. Probability in the Engineering and Informational Sciences, *35*(1), 37-50.

TensorFlow(2022). https://www.tensorflow.org/

United Nations(2019). World Population Ageing 2019: Highlights, United Nations, New York,

Wang, X., Ellul, J., & Azzopardi, G. (2020). Elderly fall detection systems: A literature survey. Frontiers in Robotics and AI, 7, 71.

Wei, H., & Kehtarnavaz, N. (2020). Simultaneous utilization of inertial and video sensing for action detection and recognition in continuous action streams. *IEEE Sensors Journal*, *20*(11), 6055-6063.

Wei, H., Chopada, P., & Kehtarnavaz, N. (2020). C-MHAD: Continuous multimodal human action dataset of simultaneous video and inertial sensing. *Sensors*, *20*(10), 2905.

Wei, H., Jafari, R., & Kehtarnavaz, N. (2019). Fusion of video and inertial sensing for deep learning–based human action recognition. *Sensors*, *19*(17), 3680.

World Health Organization(2021), https://www.who.int/news-room/fact-sheets/detail/falls

Wu, Y., Su, Y., Feng, R., Yu, N., & Zang, X. (2019). Wearable-sensor-based pre-impact fall detection system with a hierarchical classifier. *Measurement: Journal of the International Measurement Confederation*, *140*, 283–292. https://doi.org/10.1016/j.measurement.2019.04.002

Wu, Y., Su, Y., Feng, R., Yu, N., & Zang, X. (2019). Wearable-sensor-based pre-impact fall detection system with a hierarchical classifier. *Measurement: Journal of the International Measurement Confederation*, *140*, 283–292. https://doi.org/10.1016/j.measurement.2019.04.002

Xu, T., Se, H., & Liu, J. (2021). A fusion fall detection algorithm combining threshold-based method and convolutional neural network. *Microprocessors and Microsystems*, *82*(November 2020), 103828. https://doi.org/10.1016/j.micpro.2021.103828

Y. Li, K., C. Ho & M. Popescu(2012), A Microphone Array System for Automatic Fall Detection, IEEE Transactions on Biomedical Engineering, vol. 59, no. 5, pp. 1291-1301, doi: 10.1109/TBME.2012.2186449.

Yacchirema, D., de Puga, J. S., Palau, C., & Esteve, M. (2018). Fall detection system for elderly people using IoT and big data. *Procedia computer science*, *130*, 603-610.

Yhdego, H., Li, J., Morrison, S., Audette, M., Paolini, C., Sarkar, M., & Okhravi, H. (2019, April). Towards musculoskeletal simulation-aware fall injury mitigation: transfer learning with deep CNN for fall detection. In 2019 Spring Simulation Conference *(SpringSim)* (pp. 1-12). IEEE.

Zia, U., Khalil, W., Khan, S., Ahmad, I., & KHATAK, N. (2020). Towards human activity recognition for ubiquitous health care using data from awaist-mounted smartphone. *Turkish Journal of Electrical Engineering and Computer Sciences*, *28*(2), 646-663.

Bibliographic information of this paper for citing:

Al Meraikhi, Saleh & Al Rajab, Murab (2022). A Multimodal Approach of Machine and Deep Learnings to Enhance the Fall of Elderly People. *Journal of Information Technology Management*, 14 (3), 168-184. https://doi.org/ 10.22059/jitm.2022.88290