Vision Based Fall Prediction Using GCN-LSTM Model

MSc Research Project
Data Analytics

Bilge Su Erdogan
Student ID: 22196145

School of Computing
National College of Ireland

Supervisor: Mohammed Hasanuzzaman

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Bilge Su Erdogan		
Student ID:	22196145		
Programme:	MSc in Data Analytics	**Year:**	2022
Module:	Research Project		
Supervisor:	Mohammed Hasanuzzaman		
Submission Due Date:	12/12/2024		
Project Title:	…….………		
Word Count:	………5430……………………………… **Page Count**………………………12…………………….……..		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: …………Bilge Su Erdogan……………………………………………………………

Date: …12/12/2024………………………………………………………………………………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Vision Based Fall Prediction Using GCN-LSTM Model

Bilge Su Erdogan
22196145
MSCDA
National College of Ireland

Abstract

Currently, falling, which has become a nightmare especially for those living alone, significantly affects people's activities. The urgent intervention call of smart systems is very valuable in order to prevent the occurrence of negative situations after falling. For this reason, studies on fall prediction are increasing day by day. These studies are divided into two as sensor-based fall detection and vision-based fall detection. There are obstacles to sensor-based studies such as battery life, the presence of wearable technologies on people or the cost of systems placed on the ground. Ease of installation and visual-based studies that appeal to more than one person in a wide area were preferred in this study. A fall prediction that has not been done before is made with the GCN - LSTM hybrid model by calculating the distance of the head to the ground, the torso angle, the symmetric difference and the parallelism score with the results obtained from the systems that determine the joint points of the human body through images. It is aimed to contribute to the literature for the studies carried out to prevent real-time fall prediction and the negative situations that may occur afterwards.

Keywords: Fall Prediction, Vision-Based, Graf Convulsion Network, Long Short-Term Memory, OpenPose

1 Introduction

According to the data announced by the World Health Organization in 2021, 684,000 people lost their lives due to falls[1]. In a study conducted in the USA, poisoning was the most common cause of death among home accidents, followed by falls. According to the National Center of Health Statistics data, 31,400 people lost their lives due to falls in 2022. In addition to deaths, falls negatively affect human life by causing consequences such as bone fractures, internal bleeding, and brain functions affected by head impact. Falling, which is an action that not only has physical consequences but also has negative psychological consequences, causes the elderly to restrict their movements due to fear of falling.[1] Early interventions after a fall are of vital importance in reducing mortality rates and permanent damage to the human body. For this reason, in order to assist early intervention systems for falls, a high-accuracy, fast and effective prediction plays an important role in preventing serious consequences.

Thanks to the developments in the field of machine learning and deep learning, today's studies in the field of fall prediction can be divided into two as sensor-based and visual-based. Sensor-based technologies, accelerometers, estimate the fall action by using data based on pressure and vibration. Acceleration and angular velocity are tracked with wearable technologies. Different activities have different temporal angular velocity and acceleration, and by classifying these activities, fall activity is determined and a fall can be predicted by tracking the data [2]. Sensors that use pressure and vibration data are usually placed on the floor and reach a conclusion by observing the changes in this data. Considering the battery life of wearable devices and the need for the person to constantly carry them, they are not the best option for the elderly. Sensors placed on the floor are not preferred much in terms of the cost of placing them in homes or public living spaces.

In fall prediction based on visual data, the person's movements, poses and environmental interactions over time are analyzed. Certain human positions are classified as sitting, standing, lying and bending. This classification is done by considering the angles and distances of the nodes on the human bone structure. Machine learning and deep learning models are the pioneers of these studies. With the development of Neural Network models, it shows superior performance in interpreting images. Convolutional Neural Network (CNN) is used as a very efficient model in the field of object detection, image classification and segmentation [3]. Temporal analysis of sequential images can be done with Long Short Term Memory (LSTM), and Recurrent Neural Networks (RNN) support this in the context of video data. [4] Joints of the human skeleton and their relationships are analyzed with Graph Convolutional Network

1 https://www.who.int/news-room/fact-sheets/detail/falls

(GCN). The most important disadvantage of using visual-based estimation can be said to be that it cannot be used especially in bathrooms and toilets for privacy purposes, but this is an obstacle that will be solved with developing technology by cameras perceiving images as a silhouette instead of personal images.
In studies conducted in the field of fall estimation, high accuracy rate and low false prediction rate play a key role. In terms of reliability of systems and correct guidance of emergency response teams, the rate of false predictions should be quite low.
The aim of this project is to design a fall prediction system with a hybrid GCN-LSTM model that can be integrated with real-time systems and has higher accuracy and reliability than existing methods using visual data. The biggest contribution of this research is to reduce false positives with a novel approach by weighting the relationship of certain joints of the human body and their distance from the ground.
Research Question: 1) In comparison to existing methods, how can a hybrid GCN-LSTM model using visual-based data improve the reliability and accuracy of fall prediction?
2) Can a fall prediction decision mechanism by calculating the angles of certain joint points and weighting them increase reliability by reducing false positives?
The rest of the article is structured as follows. Section 2 presents the historical developments of sensor-based and vision-based studies, and a history of techniques used to detect human bone structure as a fall. Section 3 describes the methodology introduction that determines the operation of the system, the methods used for cleaning and pre-processing the data, and then for structuring the model. Section 4 presents implementation of this study, section 5 includes results and evaluation parts and finally Section 6 concludes with a summary of the study and a discussion of the future works.

2 Related Work

2.1 Wearable Sensors

A lot of studies have been conducted in the field of fall detection with the analysis of data obtained by sensors that can measure angular velocity and acceleration, which can be placed on wristbands, belts or other textile products [5]. G.L Santos et al. (2019) proposed a CNN-3B3Conv model based on the CNN model and presented a fall prediction study with accelerometers. In this study, data augmentation was applied to increase the accuracy rate. B.Saha et al (2023) conducted a prediction study with block chain technology using accelerometer, gyro and magnetometer data. By comparing K means clustering, Neural Network, Naive Bayes and Logistic Regression, the highest accuracy rate of 95% was achieved with K means clustering. H.Yhdegi et al. (2023) suggested using multi-window segmentation to speed up the performance. In this study, a model was created by using convolution-based learned features extractor and LSTM together and the F1 score was stated as 97%. J.He et al. (2013) normalized the three-axis acceleration and angular velocity data according to the range specification and mapped them to RGB bitmap. The training of the FD-CNN model they designed with this data was performed and high level of specificity and sensitivity was achieved. The fact that wearable technologies can only be used while the user is carrying them is the biggest obstacle to studies in this field. Since the target audience is especially the elderly, it is not always possible to charge them constantly or carry them on the user.

2.2 Visual Based Sensors

In studies conducted in the field of fall detection using camera images, human movements are identified and these movements are classified to detect falls. Although privacy seems to be an obstacle to using this method since the person's image is usually used, it aims to eliminate this obstacle with depth cameras. X. Kong et al. (2018) used RGB images for fall detection using the fast Fourier transform FFT. In this study, users' privacy is prioritized. R. Hasib et al (2021) applied R-CNN to extract human silhouette from computer vision based visual data and studied the performance with various machine learning algorithms by performing action classification with CNN. F. Harrou et al. (2019) considered the human body in five different parts and conducted a fall detection study on abnormalities in human movements with generalized likelihood ratio approach. Since GLR alone could not distinguish falling from similar actions such as bending or lying down, they supported their model with SVM. C. Rouiger et al (2007) conducted a detection study by observing whether there was any movement after a human fell to the ground using motion history image and change in the human shape approaches. This study may not be a solution for emergency situations as it will delay the fall decision mechanism. Approaches such as OpenPose and AlphaPose, which extract the bone structure of the human body with 2D or 3D cameras, have created a new field in the field of visually based fall detection. H.Ramirez et al. (2021) worked on fall detection results with various models using Alpha Pose for data extraction. According to the study, the highest accuracy rate is given by the Random Forest model. M. M. Hasan et al. (2019) worked on the human pose extracted by OpenPose with a hybrid model developed with RNN and LSTM. In this study, where the accuracy rate is not clearly stated, a high sensitivity rate is shared. L.C Bin et al. (2021) used images obtained with a small dataset they prepared. They published a successful study in the field of fall detection by extracting the human skeletal structure with OpenPose and using RNN-LSTM-GRU models. GRU was used to accelerate model training, but this can only be used to accelerate the training of small datasets. Since the use of RNN reduces performance, LSTM has become a more preferred model and S. Jeong et al. (2018) used the LSTM model with data obtained with 3D Microsoft Kinect cameras and proposed an innovative method in the classification of human movements and used the ST-GCN model. They aimed to automatically learn temporal and spatial patterns by modelling the human skeletal structure as a graph. In the context of classifying human movements with GCN, J. Yuan et al. [18] used this classification by strengthening it with a self-attention mechanism in temporal and spatial terms. All these studies mentioned have been a source of light for studies on falling from the past to the present. However, since there is still a lack of reliable studies to take action during real-time falling, continuous improvements are on the agenda in this area. Therefore, this study aims to eliminate the deficiencies of previous studies by presenting an innovative method that will reduce the false positive rate by taking into account angular weighted calculations.

3 Research Methodology & Design Specification

In this study, the publicly available EFPDS[2] dataset was utilized as the primary source of data. As shown in Figure 1, the images in the dataset were subjected to background cleaning and resizing processes. Then, the data processing step was completed via OpenPose and the feature extraction stage was started. In this stage, the weights were calculated and the outputs produced by OpenPose were updated. In the data modeling step, GCN and LSTM models were used to analyze temporal and spatial relationships in the data effectively. Finally, in the evaluation step, the accuracy of the fall predictions was calculated and the study was concluded.

The EFPDS dataset contains different positions of people in indoor environments. While some of the photographs do not contain people, some may contain more than one person in different positions. The dataset consists of three separate groups: train, validation and test. In total, this dataset consists of 6982 images, and there are 5023 images defined as falls.

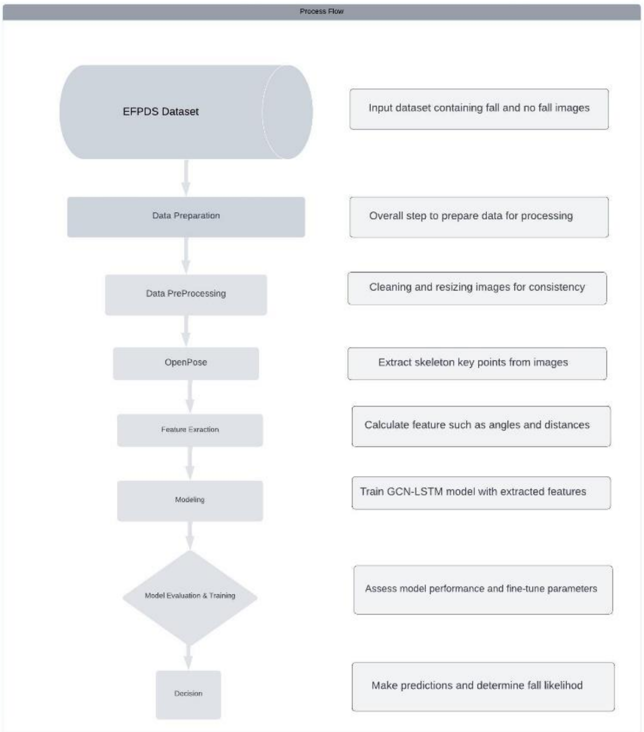

Fig. 1 Process of the research

2 https://gram.web.uah.es/data/datasets/fpds/index.html

3.1 Data Pre-processing and Processing

In order to read human poses on computers and make them meaningful, the method of making an estimation by first taking into account the bone structure of the human body and the position of the joints is frequently used. Once the bone structure of the human body and the positions of the joints are determined, a person's actions and even his intentions can be predicted based on his posture. Temporal analysis of human movements can be made with the temporal order of these joints, that is, the structures we call key points.

3.1.1 OpenPose

OpenPose is a system that estimates pose based on the bone structure of the human body. It makes pose estimation by detecting human or people from images using CNN-based deep learning architecture developed by Carnegie Mellon University. It uses Part Affinity Fields (PAF) and heat map. PAF are vector fields that determine the different parts of the human skeleton and the relationships of these parts with each other. In the context of this relationship, different human poses are determined by determining which person the joint points belong to. First, the joint points are determined at the basic level and the prediction is improved through repeated processes, resulting in a more accurate detection.

When estimating the pose for the body, OpenPose determines 25 joint points using the head, neck, shoulders, elbows, wrists, torso, hips, knees and ankles. It contributes to the analysis of human pose through the direct relationships of these 25 joints with each other. It uses more than 70 points for facial reading, allowing analysis of facial features and people's emotions. It also has a recognition mechanism for hands and is very useful in identifying hand movements by extracting 21 joint points from each hand. OpenPose BODY_25 offers options for different usage areas using COCO and MPI models. While BODY_25 and COCO models extract comprehensive body joint points, MPI has a lighter structure by estimating poses over approximately 15 joint points and can be preferred for small projects. Since the most efficient results were obtained with BODY_25, BODY_25 was preferred in this system. In this model, which also offers hand and face support, only body position is used. OpenPose takes video or images as input and produces two different outputs. First is key point data. The coordinates of the defined human pose in the x and y plane are given for each point and a confidence score is assigned. Outputs are recorded in a one-dimensional rotation type. Second is images with the graph. Each joint point defined on the image taken as input is combined with lines of different colors and recorded as a new image.

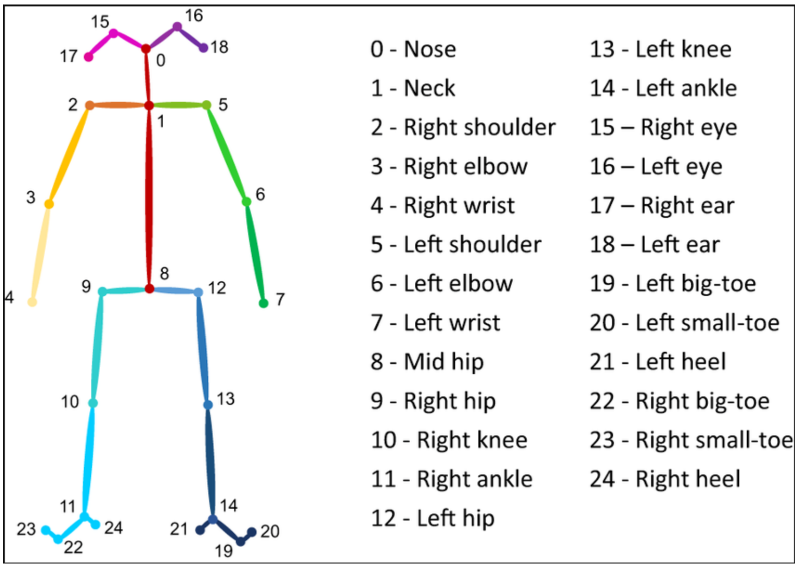

Fig. 2 OpenPose Keypoints

In order to obtain a better result compared to the original images, the data to be used as input to OpenPose, which will be used in the data processing process, has undergone pre-processing such as resizing and background filtering. The obtained images were used to produce outputs containing the spatial structures of the joint structures of the people in the images and the relationships of these structures with each other, using the OpenPose BODY_25 model. These outputs form a secure data input to be used for the GCN model and feature extraction.

3.2 Feature Extraction

Feature extraction is a very important step for the decision mechanism. In this step, the distance of the head from the ground, symmetric difference, torso angel and ground parallel score were calculated by some mathematical calculations on the keypoint data obtained with OpenPose. These calculations were carried out with the following formulations.

$$\mathrm{D_{NA}} = \mathrm{y_{neck}} - \frac{\left(\mathrm{y_{left_{ankle}}} + \mathrm{y_{right_{ankle}}}\right)}{2}$$

$\mathrm{D_{NA}}$: Distance between neck and ankles

$$\mathrm{D_{NAn}} = \mathrm{y_{nose}} - \frac{\left(\mathrm{y_{left_{ankle}}} + \mathrm{y_{right_{ankle}}}\right)}{2}$$

$\mathrm{D_{NAn}}$: Distance between nose and ankles

$$D_{NH} = y_{neck} - y_{hip}$$

$\mathrm{D_{NH}}$: Distance between neck and hip

$$D_{NHn} = y_{nose} - y_{hip}$$

D_{NHn}: Distance between nose and hip

$$\theta_{\mathrm{T}} = \arctan\left(\frac{\mathrm{y_{hip_{mid}}} - \mathrm{y_{shoulder_{mid}}}}{\mathrm{x_{hip_{mid}}} - \mathrm{x_{shoulder_{mid}}}}\right) \times \left(\frac{180}{\pi}\right)$$

θ_{T}: The angle between the midpoints of the hips and shoulders and the orientation of the torso relative to the horizontal axis

$$\mathrm{S_{diff}} = \left|\mathrm{x_{left_{shoulder}}} - \mathrm{x_{right_{shoulder}}}\right|$$

$\mathrm{S_{diff}}$: Horizontal asymmetry between the shoulders

$$G_P = \frac{\left(\left|y_{left_{shoulder}} - y_{right_{shoulder}}\right| + \left|y_{left_{hip}} - y_{right_{hip}}\right|\right)}{2}$$

G_P: The vertical misalignment of shoulders and hips, indicating the body’s tilt relative to the ground

A decision-making mechanism was developed by assigning weights according to the degree of importance to these calculated values in making the decision to fall. Accordingly, the priority weights of the distance of the head to the ground and the torso angle in the calculations were determined as 2, and the weights for the symmetrical difference and ground parallel score were determined as 1. According to these weights, the position providing 4 points was decided as a fall.

3.3 Data Modelling

3.3.1 Graph Convolutional Network

GCNs are a graph structure that emerged when the CNN model could not perform convolutional operations on non-Euclidian structures. GCNs are especially non-Euclidean graph structures; it is suitable on complex structures such as molecular structures and citation networks. With the information transmission process called message passing, a node receives information from other nodes around it, that is, from neighbouring nodes. In line with this information received, the node updates itself and learns the patterns. In this way, the information in the graph structure becomes more meaningful, and thus GCN enables the nodes to be read and made meaningful by providing communication between nodes, as well as the traditional CNN model [20].

Node structures are stored as arrays in computer terminology. A one-dimensional matrix (1) in which the corner sets are represented is represented as a 2-dimensional matrix in the adjacency matrix (2):

A ∈ R n∗n

1. V = {v1, v2, ..., vn}
2. Aij = 1 if (vi, vj) ∈ E else 0 (1)

(vi, vj) ∈ E is included in the matrix as 1 if the vertices in the graph are connected, otherwise it is 0.
In this study, the GCN model works on all connected data. Thus, the GCN learns the relationship of all points. At the same time, the important information obtained in the Feature Extraction step contributes to the training of the GCN model. The key points obtained as the OpenPose output are updated by updating the nodes and their relationships with each other thanks to the GCN model, and a more meaningful output is produced. These outputs are given as input to the LSTM model for temporal analysis.

3.3.2 Long Short-Term Memory

LSTM can be defined as recurrent neural networks for modeling data with temporal dependencies. It was published in 1997 by J.Schmidhuber (1997) et al. It provides a three-gate mechanism to learn which information will be transferred and which will be forgotten in order to learn long-term dependencies. Thus, it manages long-term and short-term information effectively. With the Door of Forgetting, it identifies unnecessary information and forgets it. New information to be stored is determined at the entrance gate, and information to be transferred to the next step is determined at the exit gate.

$$f_t = \sigma\left(W_{f[h_{\{t-1\}},x_t]} + b_f\right)$$

$$i_t = \sigma\left(W_{i[h_{\{t-1\}},x_t]} + b_i\right)$$

$$o_t = \sigma\left(W_{o[h_{\{t-1\}},x_t]} + b_o\right)$$

$$c_t = f_t \odot c_{\{t-1\}} + i_t \odot \backslash tanh\ \left(W_{c[h_{\{t-1\}},x_t]} + b_c\right)$$

$$h_t = o_t \odot \backslash tanh\ (c_t)$$

$\boldsymbol{f}_t$: Forget gate
$\boldsymbol{i_t}$: Input gate
$\boldsymbol{o_t}$: Output gate
$\boldsymbol{h_t}$: Hidden state
$\boldsymbol{c_t}$: Cell state
$\boldsymbol{x_t}$: Input at time step t

LSTM uses Loss Functions to measure the accuracy of the model and optimize the model output. For example, while mean squared error is used in regression problems, cross entropy loss is used in classification problems, such as the fall prediction we used in this study.

Mathematically, Cross Entropy Loss is shown as follows:

$$L = -\frac{1}{N}\sum_{\{i=1\}}^{N} y_i\, log(\hat{y}_i)$$

L: Loss
N: Batch Size
y_i: Actual class label (0 and 1)
$\hat{y}_i$: Predicted probability

3.4 Evaluation

The sigmoid function is generally used in binary classification problems to transform the output of a model into a probability value. In this study, the raw scores or logits obtained by LSTM and GCN models were transformed into probability values using the sigmoid activation function.

4 Implementation

The implementation of the fall prediction was implemented on the Visual Studio Code platform using Python as the primary language. Libraries such as OpenCV, Numpy, Pandas were used for OpenPose. Machine learning models including GCN and LSTM were implemented using the Torch library. GPU-requiring operations were performed on a system with a GTX1050Ti.

With a pipeline created with Python code, the images were first pre-processed as resizing and background cleaning and placed in a subfolder of the OpenPose folder. In the next step, which is extracting features from images with OpenPose, the json files containing 2D coordinates and connection keys were saved in a result folder under the openpose folder, and the jpeg files created by adding the obtained graph to the image given as input were saved in two different folders.

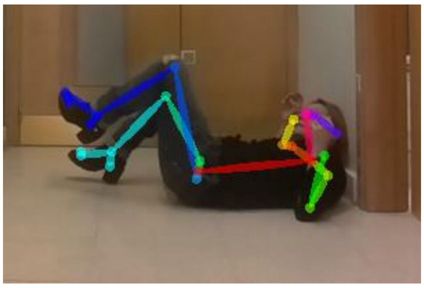

Fig. 3 OpenPose Processed Image

In the pipeline, the angle and weight calculations were made on the results obtained with OpenPose using the feature extraction function, and these calculated values were saved in another json file by producing new labels. The results with the new labels kept in another file were used to compare the effects of the results obtained with OpenPose on the fall prediction. Then, the defined GCN and LSTM models were fed with the results obtained on the pipeline and model training was performed. The learning rate was initially set to 0.001. Training was performed with combinations of 32 and 64 dimensions and different numbers of epochs and different learning rates. Hyperparameter settings were made to find the most suitable values. Finally, among the evaluation criteria, the ROC curve was used to compare the F1 score and false positive rates. The results were recorded for each experiment and are presented in the Results section in a comparative manner.

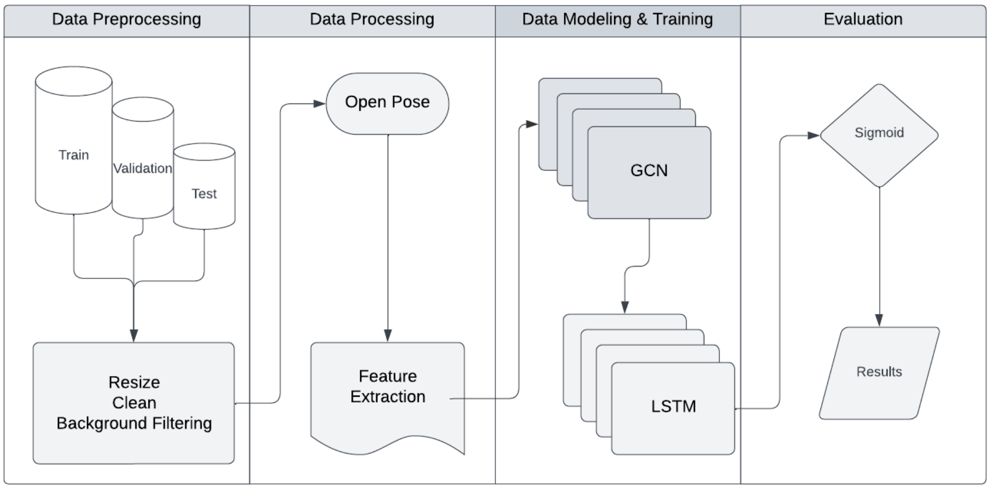

Fig. 4 Implementation Process

5 Evaluation

When Train, Validation and Test data sets are used separately, it is observed that the model performance gives the best result using 25 epochs and 32 dimensions. The F1 score is 84.15% and 84.56% accuracy rate is reached. Using a high number of epochs increases the risk of over-learning and lower epochs cannot provide full learning. In the experiments performed with the dimensions selected as 64 and 32, the fact that the 32-dimensional structure reaches a higher F1 score than the 64-dimensional structure shows that it adapts better to the limited size of the data set. However, it was observed that the 64-dimensional structure achieved a slightly higher accuracy rate, but the fact that the F1 score was slightly lower indicates that it is less reliable in false positive and false negative detections.

Epoch	Dim	F1	Accuracy	Precision	Recall
10	64	80.92%	82.92%	79.82%	82.04%
10	32	83.45%	84.02%	76.82%	91.33%
25	64	84.06%	84.97%	79.02%	89.78%
25	32	84.15%	84.56%	76.92%	92.88%
50	64	82.76%	82.92%	74.63%	92.88%
50	32	80.58%	81.83%	76.24%	85.45%

Table 1 Results with different parameters

The success rates of the GCN model and the proposed hybrid model seem to be close to each other. In this context, it is observed that the success of the GCN model in fall prediction is quite important. Although LSTM alone cannot be as successful as GCN in interpreting key points, it stands at a supportive point with temporal analysis. As a result, the proposed hybrid model has achieved more successful results than standalone models.

Model	F1	Accuracy	Precision	Recall
GCN	80.28%,	82.00%	94.20%	69.94%
LSTM	71.42%	71.24%	74.46%	68.63%
Proposed Hybrid Model GCN + LSTM	84.15%	84.56%	76.92%	92.88%

Table 2 Comparison of models

Since the low success rates observed at this stage may be due to the limited number of test data in the data set, three different test, train, validation data were combined and this rate increased to 88.14% with 150 epochs with a random separation method of 80% train and 20% test model. It can be presented as evidence of the reliability of the model by significantly reducing the false positive rate. False positive rate plays a very important role during integration with real-time applications. It has been determined that this rate has decreased from 40% to 6.8 thanks to the use of feature extraction and hybrid model.

Data Split	F1	Accuracy	Precision	Recall
Train: Train +Validation +Test 80% Test: Train +Validation +Test 20%	88.27%	88.14 %	91.53%	85.23%
Train: Train Dataset(2161 images) Test: Test Dataset(704 images)	84.06%	84.97%	79.02%	89.78%

Table 3 Results with different dataset structures

Since CPU and GPU comparison was made, it was observed that hardware features directly affected the success of the model. By keeping all parameters such as epoch number = 1 and dimension 32 constant; observing only the difference between CPU and GPU, the model's accuracy F1 score was 84.02% with GPU usage, while the same model with CPU produced 71.62% F1 score. Again, GPU usage is necessary for low-delay predictions in real-time systems.

Device	Epoch	Dim	F1	Accuracy	Precision	Recall
CUDA	1	32	83.54%	84.02%	76.55%	91.95%
CPU	1	32	71.62%	73.91%	68.86%	74.61%

Table 4 Results with GPU and CPU usage

The regular decrease in the loss values shows that the model learns by compensating for its errors in each iteration. A stable loss curve reflects that there is no overlearning or failure to learn. In addition, the closeness between training and verification shows us how successful the generalization ability of the model is, that is, how successful it is in predicting not only the training data but also the new data.

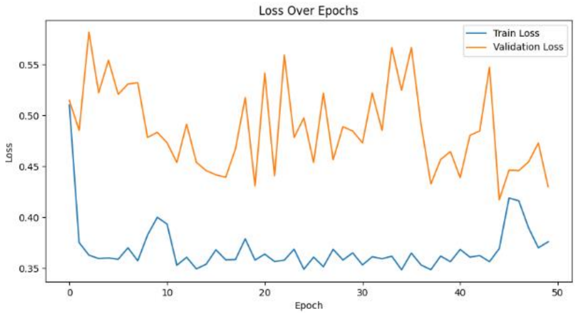

Fig. 5 Loss Values over epochs

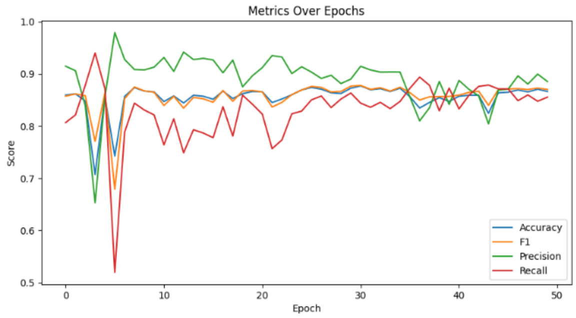

Fig. 6 Metrics over epochs

6 Conclusion & Future Work

According to the results obtained, it is possible to say that the study was completed successfully, and it was determined that it would make a great contribution to the literature for real-time fall detection applications with some improvements. When examined in detail, the important points contributing to the success of the model can be defined as the requirement of a powerful GPU and sufficient and well-detailed dataset. The layers and epoch numbers of the model are other factors affecting the success of the model. Thanks to the hybrid model, the strengths of both models were revealed in this study, and higher success was achieved compared to standalone models.

6.1 Dataset Limitations and Improvement Suggestion

The images in the dataset have limited human positions, by adding images containing more human positions, the model can be better able to recognize the difference of actions. By adding depth cameras, 3D outputs can be obtained, thus contributing to the spatial training of the model. It has been observed that people cannot be detected and no prediction can be made in some of the inputs given to OpenPose. These images can be reintroduced to the training by applying some image processing techniques on the unidentified human images. For example, the dataset can be strengthened by using techniques such as noise reduction, histogram equalization and super resolution algorithms for low-resolution images. On the other hand, if the images of human skeletons that OpenPose could not recognize are processed again using AlphaPose, it can be compared whether better results can be obtained.

6.2 Integration with Real Time Applications

In terms of integration into a real-time system, the prediction process can be applied directly on snapshots using a camera. When the GCN-LSTM model obtained with this study is integrated with real-time applications, a call for help can be made in order to prevent possible negative situations when a fall occurs. In real-time applications, the system can process video streams coming from IP cameras or CCTV in real time. After the images coming from the cameras are received in real time, skeleton points are extracted after pre-processing with this application and sent to the hybrid model. After the fall prediction is made, calls or notifications can be sent to emergency units. Areas of use can be defined as smart homes equipped with smart cameras or IOT devices, hospitals where room cameras can be used or public areas such as elderly care centers, shopping malls, parks or transportation centers.

References

[1] Sorock, G. S. (1988). Falls among the elderly: Epidemiology and prevention. American Journal of Preventive Medicine, 4(5), 282–288. https://doi.org/10.1016/S0749-3797(18)31162-0

[2] Chelli, A., & Pätzold, M. (2019). A machine learning approach for fall detection and daily living activity recognition. IEEE Access, 7, 38670–38687. https://doi.org/10.1109/ACCESS.2019.2906693

[3] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 25, 1–9.

[4] Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., & Darrell, T. (2017). Long-term recurrent convolutional networks for visual recognition and description. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(4), 677–691. https://doi.org/10.1109/TPAMI.2016.2599174

[5] Wang, S., & Wu, J. (2023). Patch-transformer network: A wearable-sensor-based fall detection method. Sensors, 23(14), 6360. https://doi.org/10.3390/s23146360

[6] Santos, G. L., Endo, P. T., Monteiro, K. H. D. C., Rocha, E. D. S., Silva, I., & Lynn, T. (2019). Accelerometer-based human fall detection using convolutional neural networks. Sensors, 19(7), 1644. https://doi.org/10.3390/s19071644

[7] Saha, B., Islam, M. S., Kamrul Riad, A., Tahora, S., Shahriar, H., & Sneha, S. (2023). BlockTheFall: Wearable device-based fall detection framework powered by machine learning and blockchain for elderly care. 2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC), 1412–1417. https://doi.org/10.1109/COMPSAC57700.2023.00216

[8] Yhdego, H., Paolini, C., & Audette, M. (2023). Toward real-time, robust wearable sensor fall detection using deep learning methods: A feasibility study. Applied Sciences, 13(8), 4988. https://doi.org/10.3390/app13084988

[9] He, J., Zhang, Z., Wang, X., & Yang, S. (2019). A low power fall sensing technology based on FD-CNN. IEEE Sensors Journal, 19(13), 5110–5118. https://doi.org/10.1109/JSEN.2019.2903482

[10] Kong, X., Meng, Z., Meng, L., & Tomiyama, H. (2018). A privacy protected fall detection IoT system for elderly persons using depth camera. 2018 International Conference on Advanced Mechatronic Systems (ICAMechS), 31–35. https://doi.org/10.1109/ICAMechS.2018.8506987

[11] Hasib, R., Khan, K. N., Yu, M., & Khan, M. S. (2021). Vision-based human posture classification and fall detection using convolutional neural network. 2021 International Conference on Artificial Intelligence (ICAI), 74–79. https://doi.org/10.1109/ICAI52203.2021.9445263

[12] Harrou, F., Zerrouki, N., Sun, Y., & Houacine, A. (2019). An integrated vision-based approach for efficient human fall detection in a home environment. IEEE Access, 7, 114966–114974. https://doi.org/10.1109/ACCESS.2019.2933216

[13] Rougier, C., Meunier, J., St-Arnaud, A., & Rousseau, J. (2007). Fall detection from human shape and motion history using video surveillance. 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07), 875–880. https://doi.org/10.1109/AINAW.2007.150

[14] Ramirez, H., Velastin, S. A., Meza, I., Fabregas, E., Makris, D., & Farias, G. (2021). Fall detection and activity recognition using human skeleton features. IEEE Access, 9, 33532–33542. https://doi.org/10.1109/ACCESS.2021.3061626

[15] Hasan, M. M., Islam, M. S., & Abdullah, S. (2019). Robust pose-based human fall detection using recurrent neural network. 2019 IEEE International Conference on Robotics, Automation, Artificial Intelligence and Internet of Things (RAAICON), 48–51. https://doi.org/10.1109/RAAICON.2019.8939861

[16] Lin, C. B., Dong, Z., Kuan, W. K., & Huang, Y. F. (2020). A framework for fall detection based on OpenPose skeleton and LSTM/GRU models. Applied Sciences, 11(1), 329. https://doi.org/10.3390/app11010329

[17] Yan, S., Xiong, Y., & Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. Proceedings of the AAAI Conference on Artificial Intelligence, 32(1), 744-752.

[18] Yuan, J., Liu, C., Liu, C., Wang, L., & Chen, Q. (2022). Real-time human falling recognition via spatial and temporal self-attention augmented graph convolutional network. 2022 IEEE International Conference on Real-Time Computing and Robotics (RCAR), 438–443. https://doi.org/10.1109/RCAR54666.2022.9824689

[19] Schmidhuber, J., & Hochreiter, S. (1997). Long short-term memory. Neural Comput, 9(8), 1735-1780.

[20] Torta, E., & Orhan, S. GCN and LSTM based Multi-Human Intention Predition.