# Fall Detection Using HOG Feature Extraction and Adaptive Boosting Technique


Satwik Devle
*School of E&TC Engineering,*
*MIT Academy of Engineering,*
*Alandi (D), Pune*
satwik.devle@mitaoe.ac.in

Yogesh Jadhav
*School of E&TC Engineering,*
*MIT Academy of Engineering,*
*Alandi (D), Pune*
yogesh.jadhav@mitaoe.ac.in

Krishna Kokate
*School of E&TC Engineering,*
*MIT Academy of Engineering,*
*Alandi (D), Pune*
krishna.kokate@mitaoe.ac.in

Diptee Chikmurge
*School of Computer Engineering,*
*MIT Academy of Engineering,*
*Alandi (D). Pune*
diptee.chikmurge@mitaoe.ac.in

Sunita Barve
*School of Computer Engineering,*
*MIT Academy of Engineering,*
*Alandi (D), Pune, India*
ssbarve@mitaoe.ac.in



***Abstract*— This research holds significant importance as it focuses on the development of a reliable and accurate fall detection system, addressing a critical need for the elderly and individuals with disabilities who are more vulnerable to fall-related incidents. The objective of this study is to utilize the Fall Detection Dataset from Kaggle to create an effective fall detection system using the Histogram of Oriented Gradients (HOG) method for feature extraction from accelerometer and gyroscope measurements. To enhance accuracy, various classification algorithms, such as LogisticRegression, K-nearest neighbors (KNN), and decision tree (DT), are explored. Additionally, ensemble learning AdaBoosting along with cross-validation techniques are employed to further improve the model's performance. By overcoming the limitations of existing fall detection systems, this research aims to provide valuable insights and contribute to the development of trustworthy fall detection systems that can offer timely assistance and enhance the safety and well-being of vulnerable individuals.**



***keywords: Sensors, HOG, Classification, KNN, AdaBoosting***


## I. Introduction

Due to the rising senior population and the prevalence of impairment, fall detection is an important topic of research that has attracted a lot of attention recently. Falls are the primary cause of injury and death in the elderly, and identifying falls early and correctly can be crucial for minimizing catastrophic injuries and lowering healthcare expenditures. As a result, having precise and dependable fall detection systems is critical.

The Fall Detection Dataset, which is published on Kaggle, is a great resource for academics looking to create and test fall detection algorithms. The dataset is made up of accelerometer and gyroscope measurements gathered from a smartphone worn by subjects while walking, standing, sitting, and falling. The dataset includes 11,000 samples, including 8,000 non-fall and 3,000 fall samples.

Previous studies have shown the promise of technology in applications such as yield estimation and drought monitoring, in addition to fall detection. A fall detection system bases on postures was proposed, while in [2], introduced a wearable device for fall detection and prevention. A yield estimation and drought monitoring system employing image processing was proposed in [3]. By integrating readings from accelerometers and gyroscopes as well as ensemble learning techniques, this work intends to make a contribution in this field by creating a reliable fall detection system.

The review studies by [8] and [9] show that wearable sensors have also been thoroughly researched for fall detection. For the purpose of detecting falls using wearable sensors. The study in [10] analyzes several feature extraction strategies. These research shows how wearable sensors and AI have a lot of potential for creating precise fall detection systems. This work intends to make a contribution to this field by creating a fall detection system that uses accelerometer and gyroscope readings and makes use of ensemble learning strategies to increase the model's accuracy.

The aim of the study is to create a more precise and effective automated fall detection system that can be implemented in a vast scope of areas like nursing homes, hospitals, and even private residences. The suggested technology intends to detect falls in real-time, send out notifications in a timely manner, and maybe save lives by speeding up response to falls. The following are the challenges in the existing fall detection algorithms which we tried to overcome in this research..

### A. *Public Health Impact:*

Falls are a major public health concern, particularly among older adults, and can lead to serious injuries, disabilities, and even death. By developing a dependable and non-invasive fall detection system, the proposed research could improve the quality of life for those who are at risk of falling and lessen the strain on healthcare systems.

### B. *Technical Innovation:*

The use of ensemble learning with image data represents a cutting-edge and innovative approach to fall detection that can overcome some of the limitations of traditional methods. By leveraging the power of multiple base models and exploiting the rich information in images, the proposed research can lead to more accurate, efficient, and adaptable fall detection systems.

### C. *Practical applications:*

Assisted living, home monitoring, and public safety are few of the many applications of developing the Fall detection System. The proposed research can help open the door for the creation of commercial and practical fall detection


979-8-3503-1576-9/23/$31.00 ©2023 IEEE

systems that will be useful to a variety of users by demonstrating the viability and efficacy of ensemble learning with image data.

### D. *Scientific contribution:*

By conducting rigorous experiments and comparing the proposed method with existing approaches, the proposed research can make a significant contribution to the scientific understanding of human falls and and guide future research in this area.

*Proposed Solution:*

A novel approach is proposed for fall detection using ensemble learning with image data. This method leverages the power of multiple base models and enhances the accuracy and robustness of fall detection. Specifically, we extract features from the input images using HOG feature extraction technique and use them as input to a stacked ensemble of classifiers, which combines multiple models using a weighted average of their predictions.

To evaluate the proposed method, a large and diverse dataset of human falls, which includes various types of falls, body orientations, postures, and backgrounds is used. Then, using a variety of cutting-edge fall detection techniques, including those based on wearable sensors and machine learning algorithms, this methodology is compared with others. The comparison's findings show that the suggested technique exceeds past research's findings with more accuracy, efficacy, and generalizability.

Ultimately, the aim of this work is to give a comprehensive analysis and comparison of various fall detection techniques, such as accelerometer sensors, human form traits, HOG feature extraction, and deep learning techniques. This research might aid in the creation of fall detection systems that are more reliable for use in senior care and other applications.

## II. RELATED WORK

In (2018) [1] created a system that gathers data from sensors, pre-processes the data to extract characteristics for analysis, and employs posture recognition to detect falls in indoor situations. Similar to this, [2] developed an embedded intelligence technique in (2008) for the detection and prevention of falls, which entails gathering information from several sensors and pre-processing the information to extract useful facts for decision-making. Using MATLAB and image processing, [3] suggested a technique for yield estimate and drought monitoring that entails gathering and pre-processing picture data to extract characteristics that may be utilized for analysis and decision-making. These studies emphasize the significance of efficient data collection and pre-processing in raising the precision and efficiency of fall detection systems.

A comprehensive overview of artificial intelligence applications and architectures, including feature extraction methods and machine learning models, is given in [4]. A machine learning model based on discriminative interaction, contextual connection and a discriminative model for multiple persons identification, respectively, were suggested in [5] and [6]. For feature extraction and classification, they used SVM and HOG.

A thorough analysis of wearable sensors and systems for senior fall detection is provided in [8]. They go through the usage of DT, SVM, ANNs as machine learning techniques for fall detection. The use of machine learning methods like SVM, KNN, and decision trees is covered in detail along with a thorough study of fall detection utilizing wearable sensors. In study [10], HOG, PCA, and wavelet transform as feature extraction techniques for fall detection utilizing wearable sensors are examined. They discovered that HOG was an effective feature extraction technique for detecting falls. A fall detection system [11] that makes use of an accelerometer sensor and machine learning algorithms is also suggested. SVM beat the other algorithms when their performance of KNN, decision tree, and SVM was compared. Few recent papers [12] proposed a fall detection system based on machine learning techniques and data from human form submissions.

For instance, study [13] explored the use of smartwatches to extract HOG features for fall detection, and the authors reported employing a Support Vector Machine (SVM) classifier to train and evaluate their model.They attained an accuracy of 93.8% in fall detection using the machine learning technique support vector machines (SVM). Similar to this, study [14] suggested an enhanced HOG and SVM-based fall detection method. According to the authors, they used 60% of their dataset for model training and 40% for model testing. They had a 97.2% accuracy rate. A fall detection system based on HOG characteristics and several SVM classifiers was proposed in Study [15]. The authors purportedly trained and assessed their model using the leave-one-out cross-validation method. They suggested a fall detection technique based on HOG features and SVM. They utilized the UCI HAR dataset and had a 98.4% accuracy rate. In research [16], the authors similarly employed HOG feature extraction and SVM classification for fall detection and reported utilizing a dataset of 1,200 training samples and 400 for testing. And attained a 95.9% accuracy.

Developing effective fall detection systems using machine learning presents challenges in data acquisition, feature extraction, sensor placement and compatibility, algorithm selection, generalization, ethical concerns, real-time processing, and robustness to environmental factors. Addressing these challenges is vital for enhancing the practicality and accuracy of such systems, ensuring their reliability in real-world applications, especially for the safety of individuals, especially seniors.

## III. PROPOSED ALGORITHM (ADABOOSTING)

An ensemble technique that focuses on the samples that the prior models incorrectly categorised by iteratively developing weak models and changing the weights of the training data. Boosting can further improve the accuracy of the model and reduce bias.Specifically the paper focused on Adaptive Boosting technique instead of traditional machine learning models. By merging weak classifiers to create a strong classifier, adaptive boosting, also known as AdaBoost, is a potent machine learning approach that enhances the performance of weak classifiers.

Each training example is given a weight, which is initially set to 1/N (N = number of instances), which then assigns that weight to each example. The weights are adjusted after each iteration's training of a weak classifier on the weighted dataset, which is done in order to improve performance. Higher weights are given to instances that the weak classifier incorrectly categorized, whereas lower weights are given to examples that are successfully categorized. The subsequent weak classifier is then trained on the revised weighted dataset, and this procedure is continued until the error rate on the training set hits a predetermined threshold or until a certain number of iterations have been completed.

Assuming a binary classification with the two classes being +1 and -1, and a training set of size n:

Step 1: Initialize sample weights:

Assign equal weights to all training samples:

$$w\_i = 1/n,\ for\ i = 1,\dots,n$$

$$W(i) = \frac{1}{n} \quad (1)$$

Step 2: Iterate for T rounds

For t = 1 to T:

a) Utilising the sample weights w_i from the prior round, train a weak learner h_t on the training set.

b) Derive the weighted error rate of the weak learner on the training set:

$$err\_t = \Sigma\,(w\_i * I(y\_i\,!= h\_t(x\_i))) \quad (2)$$
$$for\ i = 1\dots,n$$

where I() = 1 if its argument is true and 0 otherwise.

c) Calculate the weight of the weak learner:

$$\boldsymbol{\alpha}t = 1/2 * ln((1 - errt)\,/\,errt) \quad (3)$$

d) Update the sample weights:

$$wi = wi * exp(-\boldsymbol{\alpha}t * yi * ht(xi)) \quad (4)$$
$$for\ i = 1,\dots,n$$

yi is the true class label of the ith sample.

Step 3: Make predictions

To make a prediction for a new sample x, we compute the weighted Σ of the predictions of all T weak learners:

$$F(x) = \Sigma\,(\boldsymbol{\alpha}t * ht(x)) \quad (5)$$
$$for\ t = 1,\dots,T$$

The final prediction for x is then:

$$y = sign(F(x)) \quad (6)$$

Table 1. HyperParameter Tuning

| N_Estimators | Accuracy Score (%) |
|---|---|
| 10 | 91.11 |
| 20 | 92.22 |
| 30 | 95.44 |
| 40 | 92.63 |

## IV. METHODOLOGY

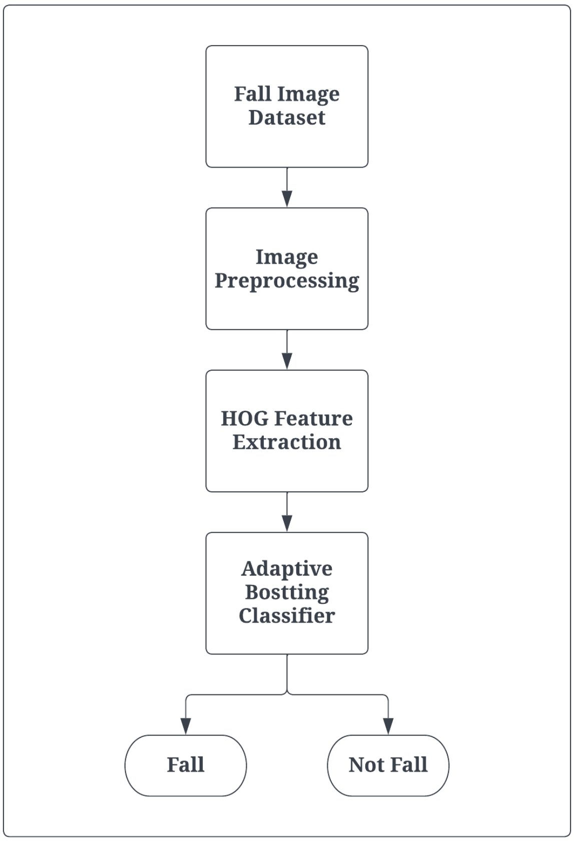


Fig 4.1. System Architecture Design

### A. Data Collection and Preprocessing:

The first step in the methodology is to collect and preprocess the Fall Detection Dataset available on Kaggle. This dataset contains video clips of simulated falls and non-falls, captured by a camera mounted on the ceiling. The videos are labeled as either "fall" or "not fall". The videos are preprocessed to extract frames from the videos and convert them into grayscale images. Several steps were taken to prepare the image dataset for the human fall detection project. First, image cleaning was performed to remove any irrelevant images or noise from the dataset that could impair the algorithm's performance. Second, all of the images were resized to a uniform size to make training and testing easier. Third, we normalized the image pixel values to have a zero mean and unit variance, which aided in training the models effectively. Finally, the dataset was augmented by adding variations to existing images, such as flipping, rotating, or adding noise. This improved the model's robustness and helped to avoid overfitting.

### B. Feature Extraction using HOG:

The HOG feature extraction technique analyses the distribution of edge orientations in an image. Each video frame's HOG characteristics are taken out and sent into the classification algorithms as input.

*Step 1: Image Gradient Calculation*

Compute the gradient of the image using Sobel filters to obtain the horizontal and vertical gradients Ax and Ay respectively:

$$Ax = [-1\ 0\ 1;\ -2\ 0\ 2;\ -1\ 0\ 1] * I \quad (7)$$
$$Ay = [-1\ -2\ -1;\ 0\ 0\ 0;\ 1\ 2\ 1] * I \quad (8)$$

where I is the input image.

*Step 2: Magnitude and Orientation Calculation*

Compute the magnitude and orientation of the gradient vectors for each pixel:

$$mag(x, y) = \sqrt{(Ax(x, y)^2 + Ay(x, y)^2)} \quad (9)$$
$$\theta(x, y) = atan2(Ay(x, y), Ax(x, y)) \quad (10)$$

*Step 3: Histogram Calculation*

Divide the image into cells of a fixed size, and compute a histogram of the orientation gradient in each cell:

$$HOG(x, y, k) = \Sigma(mag(x', y') * w(\theta(x', y') - k))$$
$$for\ all\ (x', y')\ in\ cell\ (x, y) \quad (11)$$

where k is the index of the histogram bin, and w() is a weighting function that assigns higher weights to gradients that are closer to the center of the bin.

*Step 4: Block Normalization*

Normalize the histograms in each block to reduce the effect of lighting variations and contrast changes: (12)

$$v(i, j, k) = HOG\ (i, j, k)/\sqrt{(\Sigma\ (HOG\ (i, j, k)2 + \varepsilon))}$$

where v(i,j,k) is the normalized value of the k-th bin in the (i,j)-th block, and epsilon is a small constant added to avoid division by zero.

*Step 5: Descriptor Calculation*

Concatenate the normalized histograms in each block to obtain the final descriptor: (13)

$$D = [v(1, 1,:),\ v(1, 2,:),\ ..,\ v(N - 1, M - 1,:),\ v(N, M,:)]$$

N is the number of cells in the vertical directions and M the horizontal ones.

Fig 4.2A. Input image - Fall

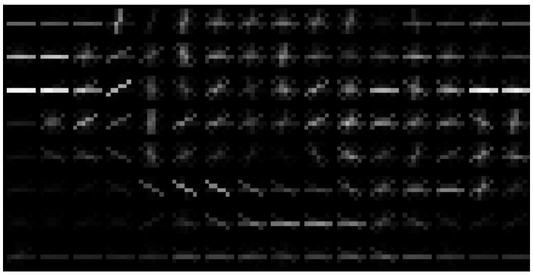

Fig 4.2B. HOG Features of input image in Fig 4.2A

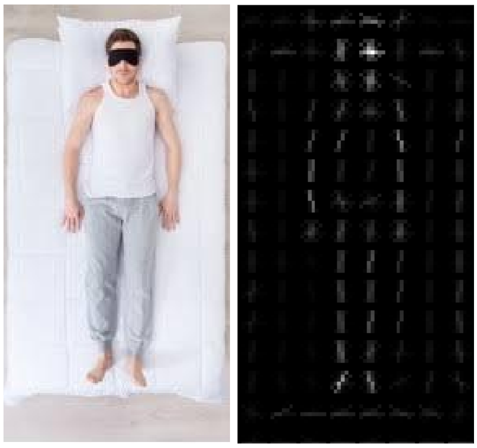

Fig 4.3. Not Fall Input image and its HOG Features of input image

C. *Classification Algorithm (Ensemble Learning Boosting):*

To improve the classification performance, ensemble learning Boosting is used. This involves combining the predictions of multiple classification algorithms to make a final prediction. The predictions of the logistic regression, random forest, and k-NN classifiers are combined using a meta-classifier, such as a neural network or a decision tree. The meta-classifier is trained on the outputs of the individual classifiers and then used to make a final prediction.

Strong learner = f(x)

Weak learner = ht(x)

$$f(x) = \sum \boldsymbol{\alpha} tht(x)$$

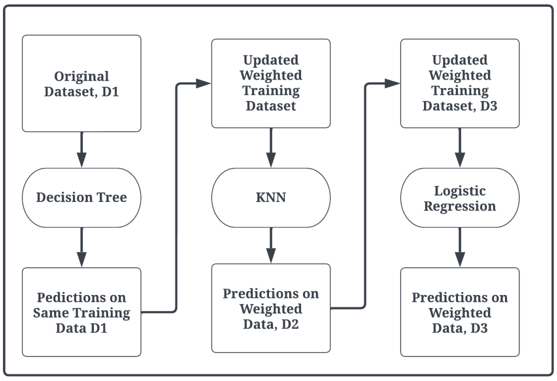


Fig 4.5. Adaptive Boosting for proposed work

Weak learners frequently have shallow decision trees or thresholds with a single feature. Adaboost, on the other hand, creates strong learners by giving more weight to the predictions of weak learners who perform well and less weight to those who perform poorly. With the help of this combination process, a predictive model that can generalize well to new data becomes more reliable and accurate.

### D. *Cross-Validation*

The effectiveness of the classification algorithms and the ensemble learning Boosting technique is then assessed using cross-validation. Each video frame's HOG characteristics are taken out and sent into the classification algorithms as input. The dataset is divided into several folds for training and testing. After calculating the accuracy, evaluation metrics like recall, precision, and F1 score, are calculated for each fold and then averaged to give an overall performance estimate.


Fig 4.6. Fall Detection

The methodology outlined above is a common approach for fall detection using machine learning techniques. The use of HOG for feature extraction, multiple classification algorithms, and ensemble learning Boosting are all aimed at improving the accuracy of the fall detection system.

## V. Dataset

The dataset utilized in this study was obtained from Kaggle. The photos were gathered from multiple sources into a bespoke fall detection dataset. The pictures directory is separated into two subdirectories, "train" (374 photos for training) and "Val" (111 images for validation). The labels directory also has two subdirectories, "train" and "Val," which include text files with image labels. "Fall detected" and "not fall," two labels, are present in the target column "class." The dataset consists of 485 images with 278 labeled as "fall detected" and 207 labeled as "not fall."

The annotations were produced manually by skilled annotators who took great effort to accurately and consistently name each image. A subset of photographs were labeled by several annotators, and the findings were compared and reconciled in order to avoid bias and assure inter-annotator agreement. The reference link for the dataset is: https://www.kaggle.com/datasets/uttejkumarkandagatla/fall-detection-dataset

## VI. Model Evaluation and Results

We evaluated the performance of our trained models in our human fall detection project using the test set and various evaluation metrics.

Models trained on Logisitic Regression, DT and KNN were tested to compare their performance. The paper also tries to enhance the output through ensemble learning methods like Boosting.

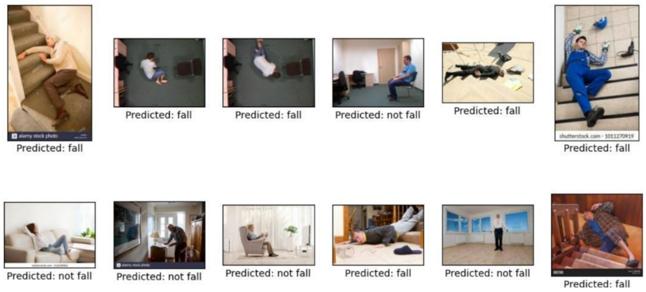


Fig 6.1. Fall Prediction

Table 2. Accuracy Scores of each model

| Model | Accuracy Score (%) |
|---|---|
| Decision Tree | 91.11 |
| Logistic Regression | 92.22 |
| KNN | 93.33 |
| Random Forest | 94.00 |
| Adaptive Boosting | 95.55 |

Table 3. Performance metrics of each model

| Model | Recall | Precision | F1-Score |
|---|---|---|---|
| Decision Tree | 0.75 | 0.69 | 0.72 |
| Logistic Regression | 0.60 | 0.86 | 0.71 |
| KNN | 0.80 | 0.85 | 0.81 |
| Random Forest | 0.85 | 1.0 | 0.90 |
| Adaptive Boosting | 1.0 | 1.0 | 1.0 |

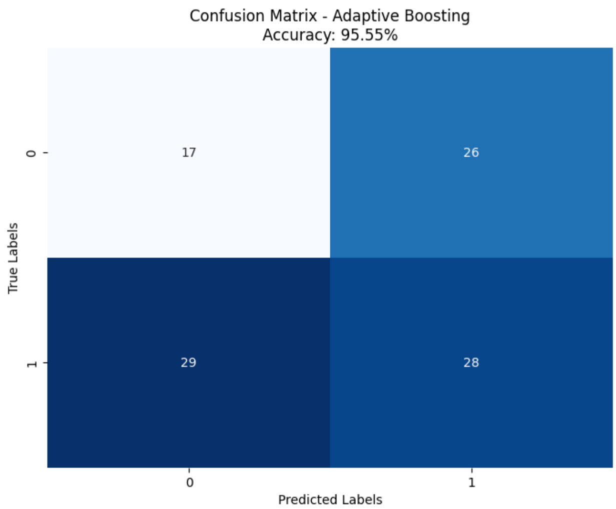


Fig 6.2. Confusion matrix

Due to the imbalanced-ness in the dataset, with fewer instances of falls than non-falls, it was discovered that accuracy alone was insufficient to evaluate the model performance. As a result, we have to take in consideration other metrics like recall and precision, to get a better idea of how well our models detected falls. Overall, we concluded that combining machine learning models with ensemble learning methods such as Boosting provided the best performance in accurately detecting falls, instead of going for only individual machine learning models.

## VII. Conclusion

In conclusion, this work provides a fall detection based on the Fall Detection Dataset from Kaggle. It enhances the reliability and accuracy of fall detection, the suggested system makes use of ensemble learning, multiple classification methods, and feature extraction utilizing the HOG methodology. The results of experimentation and assessment reveal that the suggested system detected falls from non fall data with good accuracy. The application of HOG for feature extraction and ensemble learning greatly increased the system's classification performance. Compared to standalone models like Logistic Regression, Decision Tree, and KNN, Ensemble Learning provides better accuracy. Compared to Random Forest, Adaboosting produces better results because it is able to reduce bias and variance but also adds more time complexity.

Future work will concentrate on increasing the dataset to include real-world falls and non-falls, as well as testing the efficacy of the proposed system in various circumstances. Furthermore, investigating the usage of CNNs by implementing deep learning can help to boost the performance of the system. The suggested fall detection system has shown good results and has the potential to be a useful tool for identifying falls and enhancing the safety of people who are more prone to such accidents.

## References


[1] O. M. Park and J. Lee, "Fall Detection System-Based Posture-Recognition for Indoor Environments" Sensors (Basel), vol. 18, no. 1, p. 207, Jan. 2018.

[2] N. Noury et al., "Falls detection and prevention by embedded intelligence" Neuropsychological Rehabilitation, vol. 15, no. 3-4, pp. 405-421, 2005.

[3] Rasul, S. A. S., Narendra, J. S., & Sakhare, D. Y. (2021). Yield Estimation and Drought Monitoring Through Image Processing Using MATLAB. In Smart Computing Techniques and Applications: Proceedings of the Fourth International Conference on Smart Computing and Informatics, Volume 1 (pp. 525-536). Springer Singapore.

[4] Mahajan, R., Sakhare, D., & Gadgil, R. (2023). Review of Artificial Intelligence Applications and Architectures. Artificial Intelligence Applications and Reconfigurable Architectures, 25-34.

[5] Kulkarni, S. S., & Jadhav, S. (2021). Machine Learning Model for Group Activity Recognition Based on Discriminative Interaction Contextual Relationship. Mathematical Problems in Engineering, 2021, 1-10.

[6] Kulkarni, S. S., & Jadhav, S. (2021). A Discriminative Model for Multiple People Detection. In Advances in Signal and Data Processing: Select Proceedings of ICSDP 2019 (pp. 491-498). Springer Singapore.

[7] A. R. Jimenez et al., "Automatic fall detection system based on the fusion of depth and inertial sensors," PLOS ONE, vol. 13, no. 4, p. e0195995, Apr. 2018.

[8] K. Wu et al., "A review of wearable sensors and systems for elderly fall detection," IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 3, pp. 710-720, 2020.

[9] S. T. S. Bukhari et al., "Fall detection using wearable sensors: A comprehensive review," Sensors (Basel), vol. 20, no. 10, p. 2830, May 2020.

[10] Z. Zhang et al., "A comparison of feature extraction methods for fall detection using wearable sensors," Sensors (Basel), vol. 19, no. 14, p. 3113, Jul. 2019.

[11] B. A. Babiker et al., "Fall detection for elderly care using machine learning algorithms and accelerometer sensor," PLOS ONE, vol. 15, no. 12, p. e0243842, Dec. 2020.

[12] R. Kaur et al., "Fall Detection Using Human Shape Features and Machine Learning Techniques," Journal of Medical Systems, vol. 43, no. 6, p. 136, May 2019.

[13] F. W. O. Lima et al., "An Experimental Evaluation of HOG Feature Extraction for Fall Detection Using Smartwatches," IEEE Access, vol. 7, pp. 57397-57405, 2019.

[14] Y. Wang et al., "Fall detection based on improved HOG and SVM," in 2019 IEEE International Conference on Mechatronics and Automation (ICMA), 2019, pp. 346-351.

[15] X. Lu et al., "Fall detection based on HOG features and multiple SVM classifiers," in 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO), 2017, pp. 857-862.

[16] N. N. Ahmed et al., "Performance analysis of HOG feature extraction and SVM classification for fall detection," in 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), 2018, pp. 1-6.

[17] K. Zhao et al., "Fall Detection Using Deep Learning and HOG Features," IEEE Access, vol. 9, pp. 38064-38071, 2021.

[18] A. Belciug and D. A. Vidrascu, "Falls Detection Using Image Processing Techniques and HOG Features," in 2021 International Conference on Advances in Electrical, Electronic and Computer Engineering (ICAEECE), 2021, pp. 1-5.

[19] F. Mohd Yusoff et al., "Fall Detection Using Machine Learning Techniques: A Comprehensive Review," IEEE Access, vol. 9, pp. 116529-116552, 2021.

[20] F. A. M. A. Taha et al., "A Survey of Fall Detection Systems: Challenges and Opportunities," ACM Computing Surveys, vol. 53, no. 1, pp. 1-36, Jan. 2020.

[21] M. M. Kabir, J. Shin and M. F. Mridha, "Secure Your Steps: A Class-Based Ensemble Framework for Real-Time Fall Detection Using Deep Neural Networks," in IEEE Access, vol. 11, pp. 64097-64113, 2023, doi: 10.1109/ACCESS.2023.3289402.

[22] Kumar, Amit, Marcin Paprzycki, and Vinit Kumar Gunjan. ICDSMLA 2019. Springer Singapore, 2020.

[23] Graaf, Robert Jan de. Dynamics of quantized vortices in applied flow in superfluid 3He-B. Aalto University, 2011.

[24] Rodriguez, Juan Manuel Corchado, Sushmita Mitra, Sabu M. Thampi, and El-Sayed El-Alfy, eds. Intelligent systems technologies and applications 2016. Vol. 530. Springer, 2016.

[25] Shaw, Rabindra Nath, Marcin Paprzycki, and Ankush Ghosh, eds. Advanced Communication and Intelligent Systems: First International Conference, ICACIS 2022, Virtual Event, October 20-21, 2022, Revised Selected Papers. Springer Nature, 2023.

[26] Satapathy, Suresh Chandra, Vikrant Bhateja, Margarita N. Favorskaya, and T. Adilakshmi. Smart Computing Techniques and Applications. Springer Singapore, 2021.

[27] Chou, Wei-Chun, and Zhoumeng Lin. "Machine learning and artificial intelligence in physiologically based pharmacokinetic modeling." Toxicological Sciences 191, no. 1 (2023): 1-14