Towards Safer Elderly Care: A Convolutional Neural Network Solution for Fall Detection

Kalupahana R.W[a], Dr. Maduranga M.W.P[b]

[a]Department of Information Technology, Faculty of Department, General Sir John Kotelawala Defence University, Ratmalana 10390, Colombo, Sri Lanka.
[b]Department of Electrical and Electronic Engineering, University of Sri Jayewardenepura, Nugegoda,10250, Sri Lanka.
pasanm@sjp.ac.lk , 38-bis-0001@kdu.ac.lk

ABSTRACT
As modern life becomes increasingly busy, computer vision-based monitoring systems have become essential, particularly in elderly care. This paper presents the development of a robust fall detection system using deep learning techniques, specifically a convolutional neural network (CNN) that processes RGB images to accurately distinguish between fall and non-fall events. The model is trained and validated on a dataset categorized into two classes: fall and non-fall. By utilizing convolutional and pooling layers, CNN effectively learns hierarchical representations of the input data, capturing both low-level and high-level features crucial for accurate fall detection. The key stages of this approach include data acquisition, pre-processing, and model training. The model's performance is evaluated using precision, recall, and F1-score metrics, demonstrating high accuracy, which is further enhanced through data augmentation, pre-processing, and crossvalidation techniques. A confusion matrix analysis confirms the model's effectiveness in correctly classifying instances across both classes. The system also extends its capabilities to video analysis by extracting frames at 30-second intervals, ensuring continuous and comprehensive monitoring. This research highlights the potential of deep learning to enhance safety and care for the elderly, offering a reliable solution for real-time fall detection. The findings underscore the importance of integrating advanced technologies into healthcare, paving the way for future innovations in monitoring and assistance systems.

KEYWORDS: *Fall detection, Elderly care, Computer Vision, Deep Learning, Convolutional Neural Network-(cnn), Data Preprocessing.*

INTRODUCTION
In recent years, the rapid advancement of technology has paid attention to innovative solutions in healthcare, particularly in the innovation of fall detection systems. Fall is a significant concern, especially among the elderly, as they often lead to severe injuries and a decline in quality of life. Traditional methods of fall detection, such as wearable sensors and manual monitoring, have their limitations.[1] This is where the power of deep learning, specifically Convolutional Neural Networks (CNNs), comes into play.

CNNs, a class of deep learning models, have revolutionized the field of computer vision by enabling machines to recognize and classify images with remarkable accuracy. By leveraging the capabilities of CNNs, researchers are now able to develop sophisticated fall detection systems that can analyze video feeds or sensor data in real-time, identifying falls with unprecedented precision.[2]

This paper explores the application of CNN models in fall detection systems, highlighting the advantages of deep learning techniques over conventional methods. We have learned the architecture of CNNs, their training processes, and the datasets used to enhance their performance. Furthermore, we discuss the way of detecting results with uploading video also.

The remainder of this paper is organized as follows. Section 2 provides a review of related work on computer vision-based fall detection. Section 3 dataset preparation and preprocessing of CNNbased fall detection model. Section 4 describes the evaluation model training; section 5 presents the experimental results. Finally, Section 6 discusses the key findings, implications, and future research directions.

RELATED WORKS
In recent years, there has been a growing interest in the development of advanced fall detection systems

https://doi.org/10.54389/YKYH2312

that can accurately identify and respond to falls, while minimizing false alarms [3].
One of the prominent approaches to fall detection is the use of wearable sensors, such as accelerometers and gyroscopes, integrated into devices like smartwatches or insoles.[1]These wearable systems leverage the ability to continuously monitor an individual's movement and posture, and can detect sudden changes or impacts that are characteristic of a fall [3]. For example, the study by [4]presents a wearable fall detection system that utilizes with many different sensors and powerful hardware combines with , achieving high accuracy in detecting falls with a low false alarm rate.
Complementing the wearable sensor approach, vision-based fall detection systems have also gained attention in the research community ([5]. These systems employ depth cameras or RGB-D sensors to capture visual data, which is then analyzed using machine learning algorithms to identify and classify different types of falls [6].
In addition to wearable sensors and vision-based systems, researchers have also explored the integration of multiple sensing modalities to enhance the reliability and accuracy of fall detection [7]. For instance, the study by [7]combines inertial measurement unit (IMU) data with video analysis to improve the detection and classification of falls, addressing the limitations of each individual approach.
This highlights the potential of Internet of Things (IoT) technologies in the context of elderly fall detection [8]By integrating wearable sensors, smart home devices, and cloud-based analytics, these IoTenabled systems can provide a comprehensive and connected solution for fall monitoring, prevention, and emergency response.
Beyond the technological advancements, the literature also emphasizes the importance of user acceptance and usability in the design of fall detection systems [3]. Factors such as comfort, aesthetics, and user-friendly interfaces play a crucial role in ensuring the adoption and long-term use of these systems by the elderly population.[9]
The literature review highlights the diverse approaches, from wearable sensors and vision-based systems to hybrid solutions and IoT-enabled technologies, that aim to improve the safety and well-being of the older population.
In this work, We present the development and evaluation of a novel fall detection model based on deep learning computer vision techniques. Our approach leverages a CNN architecture to analyze image data and reliably identify falls within an indoor elderly care environment. By utilizing the rich visual information captured by cameras, our system aims to provide a non-invasive and privacy-preserving solution for fall detection, and this can be used for video data also.

DATASET AND PRE-PROCESSING

This research aims to create a system to detect falls which addresses the challenges faced by caregivers in monitoring the whereabouts of elderly individuals when no direct supervision is available.

Created a dataset for model training

In the chasing of developing an effective fall detection system tailored for elderly care, our research utilized a carefully chosen dataset comprising a total of 1,424 images. The test set images should be 20% and training set images should be 80% from the total images. This dataset is split into two main classes: 'fall' and 'not fall.' The 'fall' class includes images depicting scenarios where a fall has occurred, while the 'not fall' class consists of images representing normal activities without any fall incidents like standing, running, sitting. This binary classification forms the foundation for training our machine learning model to accurately distinguish between fall and non-fall events.

The images were sourced from a combination of publicly available datasets and custom-collected data to ensure diversity in scenarios, backgrounds, and lighting conditions. This diversity is crucial for enhancing the model's robustness and generalizability across different environments and situations typically encountered in elderly care settings. A few samples of images I have tested have shown in Fig.1 , Fig.2, Fig.3 Fig.4 and Table 1 shows the number of images available in the dataset.

In the chasing of developing an effective fall detection system tailored for elderly care, our research utilized a carefully chosen dataset comprising a total of 1,424 images. The test set images should be 20% and training set images should be 80% from the total images. This dataset is split into two main classes: 'fall' and 'not fall.' The 'fall' class includes images depicting scenarios where a fall has occurred, while the 'not fall' class consists of images representing normal activities without any fall incidents like standing, running, sitting. This binary classification forms the foundation for training our machine learning model to accurately distinguish between fall and non-fall events.

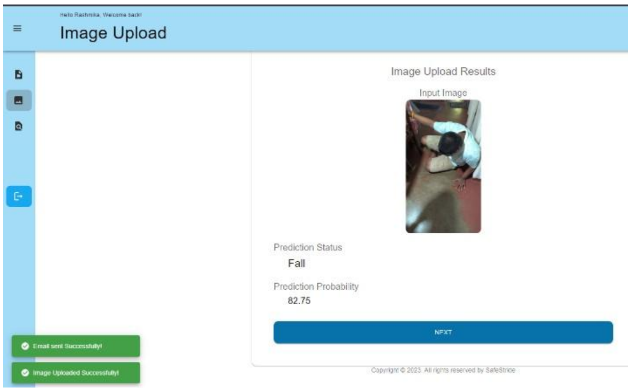

Figure 1. Fall

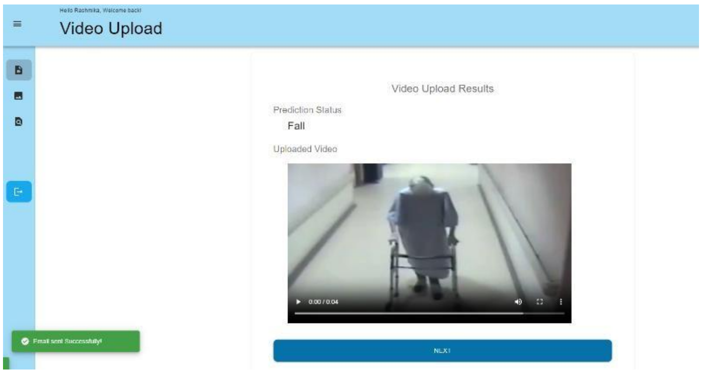

Figure 2. Video (Fall)

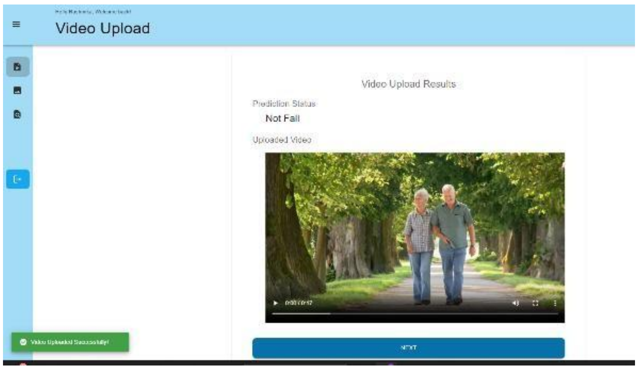

Figure 3. Not Fall

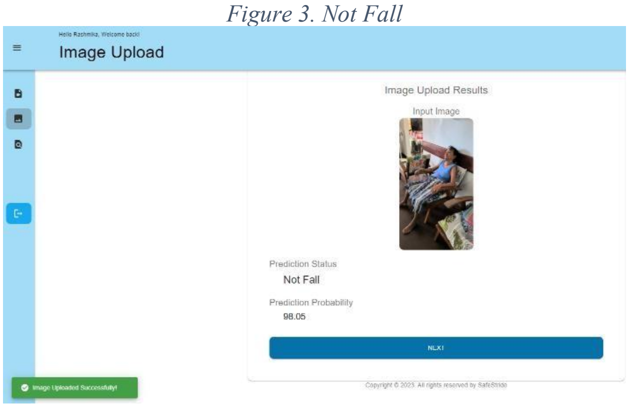

Figure 4 Video upload(Not Fall)

Table 1. Distribution of Dataset

Class	**No. of Images**

	Training set	*Test set*
Fall	653	512
Not Fall	135	124

Data Preprocessing

Data preprocessing involves procedures and strategies applied to unprocessed data before training a machine learning model. Its goal is to ensure that the data is in an appropriate format and quality for efficient model training and analysis. This requires transforming and cleaning the data.[10] The preprocessing steps applied to the training and validation datasets include rescaling,
data augmentation, resizing, selecting batch sizes, shuffling, and one-hot encoding of labels.

Initially, all images were resized to a uniform dimension to ensure consistency and compatibility with the input requirements of Convolutional Neural Network (CNN) model. Following resizing, image normalization was performed, scaling pixel values to a range between 0 and 1. This step is essential for accelerating the convergence of the model during training by maintaining uniformity in the input data.Data augmentation techniques were employed to artificially expand the dataset and introduce variability, thereby reducing the risk of overfitting. Techniques such as rotation, flipping, and zooming were applied to create multiple variations of each image, simulating different angles and perspectives. These preprocessing techniques help prepare the data for training the CNN model and improve its ability to learn and generalize from the provided images.

A total of 1424 pictures were used for the CNN model, with 1165 being used for training and 260 for verifying accuracy. The 260 images used for validation testing were chosen as 20% of the total images.

4 MODEL DEVELOPMENT AND TRAINING

4.1 Creating the CNN Model

Computer vision has dramatically enhanced thanks to convolutional neural networks, which utilize deep learning. The image illustrates the essential operation of a CNN model.

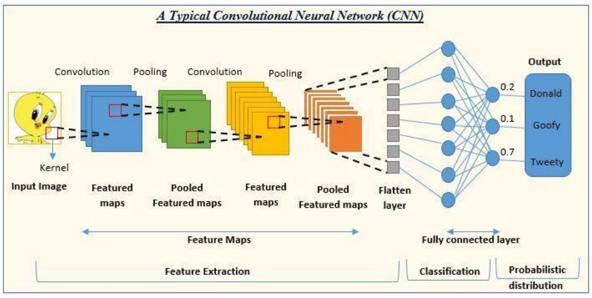

Figure 5.Basic process inside a CNN model

Input Layer:

First up, we have the input layer. It takes in RGB images sized at (255, 255, 3). This setup is standard in computer vision, where the red, green, and blue channels stack up to create a 3D tensor. This format is perfect for feeding into a convolutional neural network (CNN) [2][11]

Convolutional Layers:

The heart of the model lies in its convolutional (Conv2D) layers. These are the building blocks of any CNN. Our first Conv2D layer comes with 64 filters, each with a kernel size of 5, and uses the ReLU activation function [11][1]. ReLU, or Rectified Linear Unit, adds a dash of non-linearity, which is crucial for the model to pick up on complex patterns in the data.

When the convolutional operation kicks in, it applies a learnable filter (or kernel) to the input image. This process churns out a feature map that highlights specific visual patterns. With 64 filters and a kernel size of 5, the model is well-equipped to capture a variety of low-level features like edges, textures, and shapes, all of which are super matches fall detection [1][11][6][2].

Pooling Layers:

After the first convolutional layer, we introduce a MaxPool2D layer with a pool size of 5 [1][11][6][2] . This step is all about condensing the spatial dimensions of the feature maps. It helps trim down the model's complexity, extract more abstract features, and boost its generalization skills.

Next, the second convolutional layer steps in with 32 filters and a kernel size of 3, again using the ReLU activation function [1][11][6][2]. This layer is designed to dig deeper, capturing more complex, higher-level features from the input data, building on the groundwork laid by the first convolutional layer.

The mix of convolutional and pooling layers is a classic setup in CNN models. It allows the network to learn a hierarchical representation of the input data, moving from low-level to high-level features [1][11][6][2]. This structure is key to the model's ability to accurately spot and classify fall events from video dataModel Trainning

A crucial stage in creating a deep learning model for fall detection using the CNN architecture is model training. In this study, the training technique is carried out using a GPU (more particularly, GPU device 0) to take advantage of its processing capability and speed up the training process. The model is trained for a specified number of epochs, in this case, 40.

To minimize the stated loss function (categorical cross-entropy), the model iteratively adjusts its parameters during the training phase. This allows the model to learn from the training dataset. The Adam optimizer is used to optimize and modify the model's weights following the estimated gradients. This enables the model to gradually enhance its functionality and enhance its precision in detecting falls. The percentage of correctly categorized samples in the training dataset is known as training accuracy. The training set of data shows how accurate the model was. The model may have memorized the training data without adequately generalizing new samples, a sign of overfitting. High training accuracy, therefore, does not ensure good performance on unknown data.

The inaccuracy or difference between the model's anticipated output and the actual target values on the training dataset is represented by training loss. It shows how well the model fits the training set of data. The objective is to reduce the training loss, which measures how well the model can learn from the training data and produce reliable predictions.

A validation dataset that is separate from the training dataset is used to evaluate the model's performance at the end of each epoch's end. Researchers can learn more about the model's generalization ability and spot any overfitting or underfitting by testing it on previously unexplored data.

On the other hand, validation accuracy gauges the percentage of correctly identified samples in the validation dataset. It gives a general indication of how well the model performs on unobserved data.

A higher validation accuracy indicates a model working well while accurately identifying the samples in the validation dataset. The validation loss calculates the error or difference between the target values on the validation dataset and the model's anticipated output. It shows how effectively the Figure 3: Basic process inside a CNN model generalizes to new data. A minor validation loss indicates improved prediction accuracy on the validation dataset. After training, the model has to optimize and maintain the model.

In extending fall detection system from static images to dynamic video analysis, we have implemented frame extraction technique that allows the model to evaluate video content effectively. When a video is uploaded into the system, it is processed by capturing individual frames at 30-second intervals.

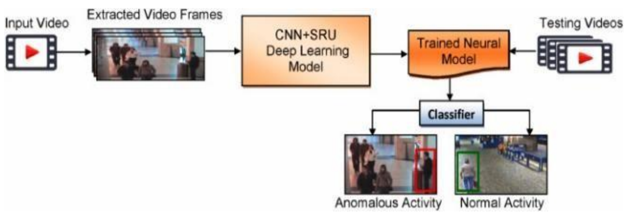

Figure 6. Extracting frames from a video

Each extracted frame is treated as an independent image and is analyzed using this trained model, which has been specifically trained to distinguish between 'fall' and 'not fall' scenarios. By leveraging the model's ability to accurately classify these frames, It can determine whether a fall event has occurred within the video.

The decision-making process involves aggregating the results from all analyzed frames. If any frame is classified as a 'fall,' the system flags the video as containing a fall incident. This method ensures that even if a fall is brief or occurs at a single point in the video, it is still detected by the system.

This integration of image-based detection into video analysis not only enhances the versatility of the model but also broadens its applicability in real-world settings, such as monitoring elderly individuals in care facilities or at home. By effectively bridging the gap between static and dynamic content, our system provides a robust solution for continuous fall monitoring.

RESULTS AND DISSCUSSION

This section presents the outcomes and implications of the research on fall detection using deep learning techniques. Here, outline the performance metrics achieved by CNN model on both the training and validation datasets, along with a thorough analysis of the results.
To fully understand the model training process and its impact on the performance of fall detection CNN model, We conducted a detailed analysis of the model's training history graph. This analysis helps to see how the model's accuracy and loss evolved over time, providing insights into its learning behavior and stability.
By examining these metrics, can assess how well the model generalizes to new, unseen data and identify any areas for improvement. This comprehensive evaluation is crucial for refining approach and ensuring the model's effectiveness in real-world applications.

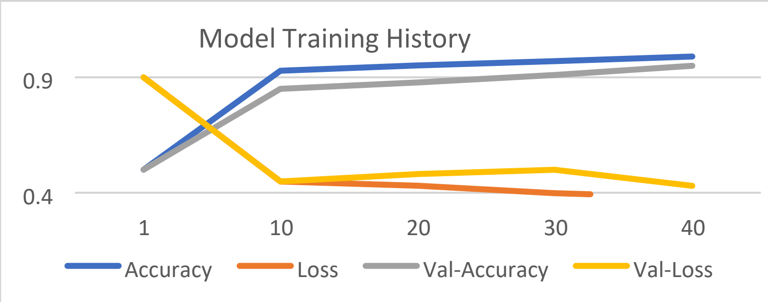

Figure 7.Model Training History

Figure 7 indicates the predicted behavior of the models produced, allowing them to evaluate whether the models should be rejected or accepted. The blue and red lines show the accuracy and loss of the training set, respectively. The green and yellow lines show the validation accuracy and validation loss, respectively.
Increasing the number of epochs can improve accuracy by allowing the model to make more weight and bias updates, leading to better pattern recognition and learning from the training data. This model was trained using 40 epochs, and both accuracies increased as the number of epochs increased, while both losses decreased.

Table 2. Model Summary

Layer (type)	Output shape	Param #
conv2d_3(Conv2D)	(None, 253, 253, 32)	896
max_pooling2d_3 (MaxPooling2d)	(None, 126, 126, 32)	0
conv2d_4(Conv2D)	(None, 124, 124, 64)	18,496
max_pooling2d_4 (MaxPooling2d)	(None, 62, 62, 64)	0
conv2d_5(Conv2D)	(None, 60, 60, 128)	73,856
max_pooling2d_5 (MaxPooling2d)	(None, 30, 30, 128)	0
flatten_1(Flatten)	(None, 115200)	0
dense_3(Dense)	(None, 128)	14,745,728
dense_4(Dense)	(None, 64)	8,256
dense_5(Dense)	(None, 2)	130
Total params: 14,847,362		

Trainable Params: 14,847,362
Non-trainable Params: 0

The model summary offers a clear snapshot of the architecture and structure of the trained CNN model. It details the various layers, including their types and output shapes, providing insight into how the model processes information. Additionally, it lists the number of parameters associated with each layer, highlighting the model's capacity and complexity.

In total, the model comprises 14,847,362 parameters, all of which are trainable. There are no nontrainable parameters, meaning every parameter is adjusted during training to help the model learn and make accurate predictions based on the input data.

With its numerous convolutional and pooling layers, the model is adept at recognizing complex patterns and extracting crucial information from input images since it has many convolutional, pooling and dense layers. This layered structure allows the model to effectively identify features necessary for distinguishing between different scenarios, such as falls and non-falls, enhancing its predictive accuracy and reliability.

Table 3. Classification Report

	precision	recall	f1-score	support
0	1.00	0.99	1.00	653
1	0.99	1.00	0.99	511
accuracy			0.99	1164
macro avg	0.99	1.0	0.99	1164
weighted avg	0.99	0.99	0.99	1164

A classification report is an invaluable tool for evaluating the performance of a CNN model in a classification task. It provides a detailed analysis of various metrics for each class in the dataset, including precision, recall, F1-score, and support.

Precision measures the accuracy of positive predictions for a specific class. It indicates how well the model can classify instances belonging to that class without generating false positives. A high precision score reflects a low rate of false positives. Recall, also known as sensitivity or the true positive rate, assesses the model's ability to identify positive examples of a class. It shows the model's effectiveness in detecting relevant events without producing false negatives. A high recall score signifies a low percentage of false negatives.

Table 4 .Confusion Matrix

Fall	647 (99.1%)	6(0.9%)
Not fall	7(1.5%)	504(98.5%)
	Fall	Not Fall

According to the matrix:

Fall: Out of 653 instances, 647 (99.1%) were correctly identified as falls, showing strong proficiency in detecting falls.

Not Fall: All 511 instances were correctly identified as not falls, achieving a perfect score.

A confusion matrix is a useful tool that compares the predicted labels for each class in a dataset with the actual labels, providing a detailed review of model performance.

True Positive (TP): The model correctly predicted the positive instances of a class.

Negative (TN): The model correctly predicted the negative instances of a class.

False Positive (FP): The model incorrectly predicted positive instances, also known as Type I error or false alarms.

False Negative (FN): The model incorrectly predicted negative instances, also known as Type II error or missed detections.

CONCLUSION AND FURTHER WORKS

This study successfully developed a fall detection system using deep learning methods, specifically a Convolutional Neural Network (CNN), to monitor elderly individuals in indoor environments. The system addresses the challenge of remotely monitoring elderly individuals when direct supervision is not available. By achieving high accuracy and reliability, the proposed solution enhances safety and significantly reduces response times in case of emergencies. The findings emphasize the importance of integrating advanced technologies into healthcare, setting the stage for future innovations in monitoring systems.

The implementation of this system in elderly care settings can greatly improve safety by providing real-time fall detection. This technology not only reduces emergency response times but also offers peace of mind to caregivers and family members. The model's high accuracy ensures its reliability, making it a valuable tool for monitoring and assistance. When a fall is detected, an alert is automatically sent to the caregiver, ensuring timely intervention.

For future work, the development of a web application for caregivers is planned, which will provide real-time alerts in case of a detected fall and offer updates on fall detection history. Additionally, future research could explore indoor localization for elderly individuals using an Indoor Positioning System (IPS). The goal is to develop this system into a comprehensive product for use in elderly care homes, hospitals, and residential complexes.

REFERENCES

[1] C. P. Shewell and B. S. Hons, "Leveraging Machine-Vision for Activity Recognition Utilising Indoor Localisation To Support," 2023.

[2] S. Sherin, P. M. T. Student, and A. J. Assistant, "Human Fall Detection using Convolutional Neural Network," *Int. J. Eng. Res. Technol.*, vol. 8, no. 6, pp. 1368–1372, 2019, [Online]. Available: www.ijert.org

[3] R. Igual, C. Medrano, and I. Plaza, "Challenges, issues and trends in fall detection systems," *Biomed. Eng. Online*, vol. 12, no. 1, 2013, doi: 10.1186/1475-925X-12-66.

[4] L. Árvai, "Application of Smartwatches in Elderly Care with Indoor Localization Functionality," *Int. J. Interact. Mob. Technol.*, vol. 15, no. 5, pp. 174–186, 2021, doi: 10.3991/ijim.v15i05.20435.

[5] D. Asmar, A. Fakih, and J. Zelek, "COMPUTER VISION FOR ASSISTIVE INDOOR LOCALIZATION".

[6] H. Kwon *et al.*, "Indoor Localization and Multi-person Tracking Using Privacy Preserving Distributed Camera Network with Edge Computing," pp. 1–13, 2023, [Online]. Available: http://arxiv.org/abs/2305.05062

[7] T. M. Le, V. T. Ly, and S. V. T. Dao, "A Feature Selection Approach for Fall Detection Using Various Machine Learning Classifiers," *IEEE Access*, vol. 9, no. August, pp. 115895–115908, 2021, doi: 10.1109/ACCESS.2021.3105581.

[8] M. Oudah, A. Al-Naji, and J. Chahl, "Computer Vision for Elderly Care Based on Deep Learning CNN and SVM," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1105, no. 1, p. 012070, 2021, doi: 10.1088/1757-899x/1105/1/012070.

[9] J. Schwickert, L., Becker, C., Lindemann, U., Maréchal, C., Bourke, A., Chiari, L., ... & Klenk, "Fall detection with body-worn sensors: a systematic review.," 2013.

[10] M. Inayathulla, P. H. Kiran, M. C. Sri, and M. Deepika, "A Heuristic Model for Predicting Human Fall Detection using Machine Learning Techniques," *Int. J. Recent Technol. Eng.*, vol. 9, no. 2, pp. 119–121, 2020, doi: 10.35940/ijrte.b3180.079220.

[11] E. Alam, A. Sufian, P. Dutta, and M. Leo, "Real-Time Human Fall Detection Using a Lightweight Pose Estimation Technique," *Commun. Comput. Inf. Sci.*, vol. 1956 CCIS, pp. 30– 40, 2024, doi: 10.1007/978-3-031-48879-5_3.