Tackling Visual Illumination Variations in Fall Detection for Healthcare Applications

CHIBUIKE IKECHUKWU, Miracle and WANG, Jing <http://orcid.org/0000-0002-5418-0217>

Tackling Visual Illumination Variations in Fall Detection for Healthcare Applications

1st Miracle Chibuike Ikechukwu
Department of Computing
Sheffield Hallam University
Sheffield, UK
mcikechukwu01@gmail.com

2nd Jing Wang
Department of Computing
Sheffield Hallam University
Sheffield, UK
jing.wang@shu.ac.uk

***Abstract*—This paper presents an innovative approach for fall detection, a significant concern in elder care, using vision-based techniques and video analysis. By employing and comparing supervised machine learning algorithms for recognising falls, The paper examines the impact of different environmental conditions on the fall detection system, focusing on illumination, an aspect previously overlooked in the field. The study introduces a vision-based fall detection method using Human Pose Estimation (HPE) models, specifically MoveNet, for feature extraction from human gestures and temporal moving features. Selected machine learning algorithms and neural network models are then trained and compared using these features to recognise video events such as falls and non-falls. The presented results show promising 70.6% accuracy and real-time model efficiency. This study's findings hold significant potential for enhancing timely fall detection in real-world scenarios.**

***Index Terms*—Human pose estimation, machine learning, fall detection**

I. Introduction

A. Background

"Falls" refers to any incident or circumstance that causes an individual to come to rest on the ground unintentionally. This can happen for various reasons, including environmental and physiological issues. The risk or likelihood of experiencing major falls tends to increase significantly for older adults. This increased susceptibility is largely due to various natural and biological factors intricately associated with the ageing process, such as decreased physical strength, balance issues, and cognitive impairments. When a fall occurs, the severity of the outcome is often compounded by delayed medical aid and the prolonged period of time the individual might lie on the floor due to the fall. This delay in receiving appropriate medical attention can lead to increased physical complications, including severe injuries, and psychological consequences, such as fear of falling or anxiety, which can subsequently reduce the individual's quality of life [1].

To prevent falls and provide a better independent living environment for older adults, healthcare providers and researchers pay great attention to computer vision and machine learning technologies, such as human gesture recognition, to detect and prevent falls. However, computer vision systems often face several challenges. These include variations in lighting conditions, different home environments, and the need for constant camera calibration. Privacy concerns are also significant, as surveillance involves continuous monitoring, which can be seen as intrusive. Furthermore, computer vision systems can sometimes struggle to distinguish between actual falls and daily activities such as sitting or bending, leading to false alarms. Therefore, while computer vision holds great potential for fall detection, these challenges must be addressed effectively to ensure successful implementation.

This paper primarily focuses on exploring new solutions for fall detection using vision-based techniques to analyse video data. This research used a supervised machine learning algorithm to detect falls. The main focus is understanding the model's performance across occlusion, illumination, and lack of illumination. The research also explores fall detection at varying illumination levels to understand the model's behaviour across these levels. The goal is to enhance prompt detection of falls in real-world challenging scenarios for computer vision and machine learning tasks.

This paper presents a vision-based fall detection technique primarily using video analysis. It begins by exploring a human skeleton model for feature extraction, accomplished through Human Pose Estimation (HPE) models, specifically MoveNet, for human stick model detection. Although other research has used HPE models like OpenPose [2] and AlphaPose [3] for fall detection, none, to our knowledge, have used MoveNet with its superior efficiency and accuracy.

Following feature extraction, classic machine learning algorithms such as Support Vector Machine (SVM) and popular neural network models such as Long Short-Term Memory (LSTM) were trained using the skeleton model feature spaces.

Lastly, the paper examines the impact of various environmental conditions on a vision-based fall detection system's outcome. It specifically assesses conditions such as occlusion and illumination, emphasising the latter. While other research has explored occlusion in fall detection, this paper highlights that the aspect of illumination has been largely overlooked.

The rest of the paper is organised as follows: A brief literature review of vision-based fall detection is presented in Section II. Section III will explore the proposed HPE model for human skeleton visual feature extraction. Section IV will then introduce the system implementation for machine learning-based human gesture recognition and fall event detection.

Section V will include a series of tests to evaluate system performance. The final section will summarise our work and suggest potential future directions.

II. Review on Vision-based Fall Detection

Generally, healthcare applications use sensors such as visual, geometric, and accelerometers from cameras or smart watches for fall detection. However, due to the rising adoption and easy deployment of cameras and video systems in healthcare smart home settings, more and more applications focus on computer vision-based approaches for real-world applications. It has been reported that with improvements in noninvasive imaging devices and cameras, the quality of images and video data for monitoring and user activity analysis has significantly improved [4].

Traditionally, vision-based fall detection is divided into four processing stages: data collection, feature extraction, feature selection, and learning/inference. Based on this four-stage framework, many previous works have explored a vision-based methodology for detecting falls using varying machine learning techniques for learning and inference and different feature extraction processes for capturing the key movement of human gestures.

For example, Charfi *et al.* [5] applied a SVM classifier to selected features obtained from tracking the silhouette of a human body to detect falls. The work used videos captured from a single camera, extracting features like projection histogram, bounding box-based height and width of human body, and human body trajectory and orientation. The features were transformed using well known Wavelet transform, Fourier transform, Second derivative for acceleration and First derivative for velocity to detect falls.

In a similar fashion, Zerrouki and Houacine [6] used SVM and the Hidden Markov Model (HMM) for the classification of fall and non-fall activities. The authors applied the curvelet transform to identify human posture by using curvelet coefficients obtained from background subtraction of images, as attributes of the human body. Calculated area ratios are also added to complement the position recognition drawback of the rotation invariant curvelet coefficients, and the feature vector reduced for posture classification and finally classified into fall and non-fall activities through the Hidden Markov Model.

In order to address issues of occlusion between the human and the camera associated with a single view, Thome *et al.* [7] used two camera video sequences and modelled a multi-view fall detection system. Their work proposed a posture classification fusion unit based on the fuzzy logic decisions. The fusion unit combines the output from each camera to generate a multi-view video sequence. Their proposed system identifies individuals in multi-view videos by subtracting the background using a Stauffer mixture of Gaussian models. It then identifies the silhouette and conducts a static analysis of the body pose. Finally, a Layered Hidden Markov Model (LHMM) is applied to classify the associated event.

Apart from using classic machine learning algorithms such as SVM and HMM. Some other methods use dynamic shape changes based on human shapes. For example, Rougier *et al.* [8] proposed a method of identifying falls by examining the distortion in human shape from a video sequence. The work tracked the silhouette of the human body along sequences of video by comparing consecutive edge points from the silhouette of a person in order to match and evaluate the shape distortion of the silhouette by use of shape analysis method such as the Mean-Shift algorithms and the Procrustes distance for shape comparison. Finally the analysed shapes are passed into a Gaussian Mixture Model(GMM) for fall or non-fall activity classification based on the distortion in the shapes.

A lot of the aforementioned works explored the classification of activities in vision-based fall detection using statistical or threshold-based classification algorithms. More and more recent research used large data-driven solutions and neural network algorithms due to the fast development of deep learning. Lu *et al.* [9] used a combination of 3-dimensional Convolutional Neural Network(CNN) and Long Short Term Memory(LSTM) to detect falls on video kinematic data. The proposed system takes an input of video sequence into a 3-dimensional CNN to extract both temporal and spatial data. It combines the visual attention of LSTM to locate and extract motion information and the region of fall in the video. The system achieved an average accuracy of 99.73%.

However, despite these advancements, there are still several uncovered knowledge gaps. Most of the existing methods face challenges with occlusion and variations in lighting conditions. Also, most current models have been trained and tested under ideal conditions, which might not always be the case in real-world scenarios. The aspect of illumination, which could affect the performance of the fall detection system, has been largely overlooked in current research.

There is also a need for more efficient and accurate models for human pose estimation, as the current methods may not meet the efficiency and accuracy standards required for real-world application in healthcare. Those gaps will be highlighted in the rest of the paper. We will keep exploring the performance of popular methods under real-world situations and evaluate their performance for the healthcare and independent living of older adults.

III. Skeleton-based Human Pose Estimation

Human Pose Estimation (HPE) detects a pose on the human body by identifying and extracting key-points from the human body in the input image or video sequence, these key-points are used to estimate the pose correctness and identify the pose. The number of extracted key-points varies across different approaches. However, 17 key-points systems are commonly used and detected by these models.

In the paper, we are using MoveNet [10] for HPE, which was introduced in 2021 as part of the TensorFlow framework. MoveNet uses bottom-up estimation model consisting of a set of prediction “heads” and a feature extractor. This pose estimation model start to detect an object by applying a predefined networks of key-point estimation to identify human body’s location, orientation, centre point and size/scale changes. The

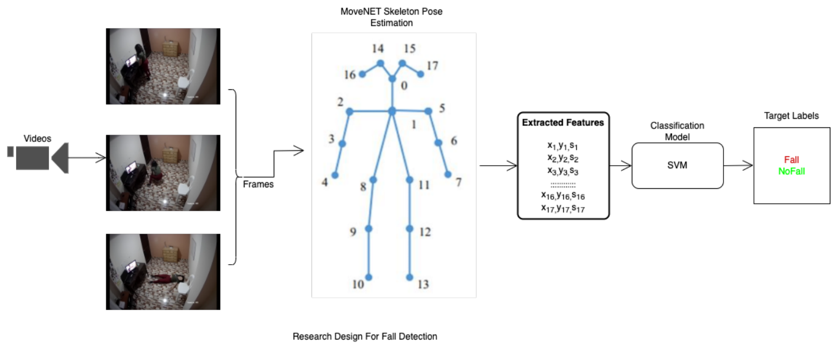


Fig. 1: System framework of the fall detection system based on human post and SVM

prediction "heads" then uses heat maps from its four feature spaces:

- **Person centre**: This is a mechanism that predicts the geometric centre of person instances. It uses shape features to accurately pinpoint the centre of each individual within the frame.
- **Key-point regression field**: This map predicts the full set of key-points for a person. It is particularly useful for grouping key-points into instances. By using this tool, we can identify the various points of a person's body and connect them to form a comprehensive picture of the person's posture and position.
- **Person key-point**: This map predicts the location of all key-points, independent of person instances. It provides a visual representation of the key-points of a person, irrespective of the individual instances. This is especially useful when you want to focus on the key-points without getting influenced by the individual instances.
- **2D per-key-point offset filed**: This map predicts local offsets from each output feature map pixel to the precise sub-pixel location of each key-point. It helps to increase the accuracy of key-point location prediction, allowing for more precise body movement tracking.

At last, the algorithm uses feature map output from the feature extractor and MobileNetV2 [11] neural network to extract distinctive features for classification, semantic segmentation and identification of objects.

MoveNet is trained on the COCO dataset and a proprietary dataset used exclusively by Google. The COCO dataset is a standard benchmark for object detection tasks that provides a wide variety of different scenes and scales. This allows MoveNet to learn from a diverse range of data, enhancing its ability to understand and interpret movement. In our work, we used the pre-trained model without modification at any level of transfer learning.

IV. Fall Detection Algorithm Implementation

In this paper we propose a system that implements a human skeleton approach to detect falls. This approach extracts joint information from frames of video sequence by applying a HPE model on the person in the video sequence. The human is tracked frame by frame with the identified key points. The extracted features classified into fall and non-fall activities using SVM and LSTM. The Skeleton-based Human Pose Estimation Model used for this research is MoveNet and is based on its comparative performance in identifying key-points relative to other HPE models such as OpenPose and PoseNet. The system framework mentioned above are illustrated in the Fig. 1

A. Dataset

In this research, the dataset used was the CAUCAFall [12] dataset that was collected by Universidad del Cauca and Universidad de la Amazonia. The Dataset consists of videos and images of several activities performed by 10 subjects. It was chosen for this project based on the quality of data and its usefulness for evaluation in an uncontrolled environment and variety of light and illumination levels as data contains videos and images from varying light conditions like natural light, artificial light and no light, when compared to other publicly available datasets.

B. Key-point Extraction

To extract key-points, 2D RGB videos are fed into MoveNet to detect the person in each frame and extract a skeletal representation of the joints on the human subject. MoveNet uses a calculation of heat map on the image frame to identify and localise the human key-points on the video frames. The result of the calculation is a 2D representation of the identified key points and a confidence score for each key-point. The pose estimation process using MoveNet outputs 17 key points per frame of the RGB image. The 17 key points are on the nose, left/right eyes, ears, shoulder, elbows, wrists, hips, knees, and ankles.

C. Data Cleaning

The data extracted from the RGB image in the form of key-points are used to prepare our dataset. The dataset is a representation of features of the human subject as derived from the key points of the human skeleton on each frame from the videos.

The extracted keypoint from the keypoint extraction process is a list element consisting of the x-coordinate, y-coordinate and the confidence score for each identified keypoint. To

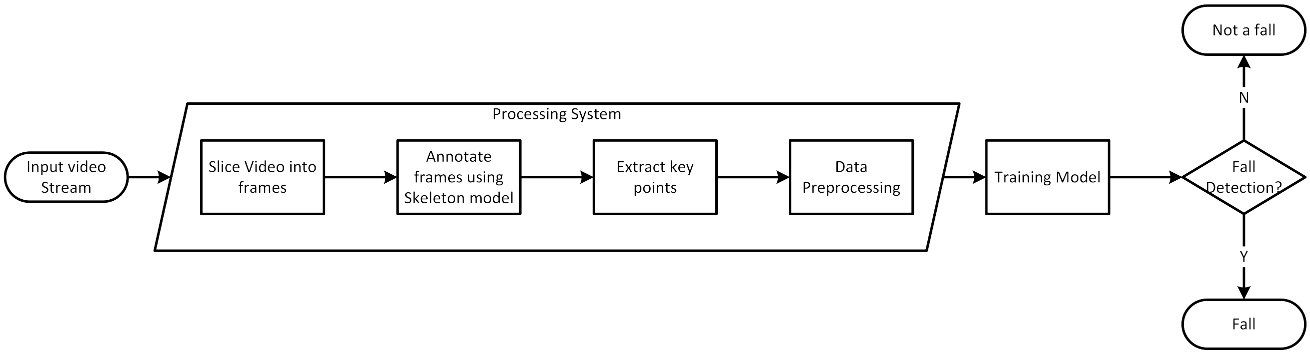


Fig. 2: System framework of Inference Pipeline

prepare this data for training, some preprocessing are carried out which includes:

- Slicing and extracting x and y coordinates from each key-point
- Remove empty data placeholders caused by failed frame-by-frame detection
- Reshape the data format for learning algorithms

D. Model Training

In the process of training our model for detecting falls, we decided to use the SVM algorithm due to its proven effectiveness when dealing with high-dimensional spaces. Additionally, SVM's compatibility with binary classification problems made it a natural choice for our fall detection project. Our approach involved feeding the cleaned dataset into the SVM on a frame-by-frame basis. Each individual frame was treated as a data sample, each tagged with either a "fall" or "non-fall" label, depending on the occurrence of a fall event. A detailed evaluation of this model's performance will be presented in Section 5.

However, one inherent drawback of the SVM is its inability to account for the temporal relationship between different frames. This lack of consideration for the sequence of events could potentially impact the accuracy of fall detection. Hence, to address this issue, we also implemented neural network algorithms such as Long Short-Term Memory (LSTM). LSTM has the capability to enhance the correlation between different frames by effectively recognising patterns over time. We sent groups of 10-30 frames as a single batch to the LSTM. This approach allowed the LSTM to analyse a temporal sequence and determine if a fall event occurred within that specific batch of frames. In the test, we used two different LSTMs containing different hidden layers to test the impact of shallow and deep networks.

E. Inference Pipeline

For this research, the implemented inference pipeline consists of some processes in the training pipeline being fed into the trained model for classification or prediction. The entire inference process runs from the input video stream, slicing video into frames, annotating frames using a skeleton model, extracting key points, and pre-processing data into the trained model where prediction is made. A workflow diagram of the steps in our inference pipeline and an explanation of some of the steps are detailed in Fig. 2.

In the figure, the trained model takes in features which are pre-processed key points from frames of unseen videos and calculates the probabilities of the possible outcomes which are 0's for no falls and 1's for falls. Our priority is to detect falls in each frame, and consequently, in the video, we set our model's fall probability threshold to 0.7, ensuring that frames with a probability greater than 0.7 are outputted as falls while frames with a probability lesser than 0.7 are outputted as no falls.

We also embedded the bounding box for location. During the inference phase, a bounding box was drawn to track the human object on each frame. Frames where fall is predicted and detected were marked with red bounding box over the human object while the human object in frames predicted no fall, were marked with green bounding box.

At the end, the frames that have been predicted and classified into fall or no fall are rendered as a video output with frames appropriately labelled fall as predicted by our model or no fall as otherwise predicted. Fig. 3 is a visual representation of an output from our inference.

V. Experiment and Performance Analysis

In the course of our experiment, our primary focus is to explore the variations in illumination and their subsequent impact on the overall performance of the system. It is a well-established fact that other image variations, including changes

Fig. 3: System framework of the fall detection system based on human post and SVM

in the view-point and the presence of background noise, also significantly contribute to the accuracy and efficiency of the system. These factors, while integral to the system's overall functioning, can complicate the data interpretation and performance analysis process. Therefore, for the purpose of this paper, we have decided to concentrate and simplify the scenarios that involve changes in lighting conditions. By doing so, we aim to provide a more focused analysis of how lighting conditions can affect system performance without the added complexity of other image variations.

A. Experiment Setup

This project and its experiments were entirely developed and performed in Python and made use of several Python libraries, including Glob, OpenCV, TensorFlow, NumPy, Pandas, and Imageio.

The test data were divided into four groups: general, light, no light, and occlusion. These experiments were conducted using the 30 videos for evaluation from a total number of 70 training videos to observe the performance of our model and the data characteristics across the different data groups:

- **General Data** refers to all the data used for testing our model, which is all 30 videos used for evaluation. These 30 videos consist of videos with light, with occluded objects and without light. Of these 30 videos, 5944 frames were extracted in total.
- **No Light Data** are all video data captured without light at an illumination of 0 lux. This dataset consists of 9 videos with 1763.
- **Light Data** refers to videos captured with either artificial or natural lighting. These lightings had varying levels of illumination, and this dataset was made up of 11 videos with 2186 frames extracted.
- **Occlusion Data** are video data captured with some occluded objects like chairs in sight irrespective of the illumination level. This dataset is made up of 10 videos and 1995 frames extracted from the videos.

B. Results and Analysis

Tab. I demonstrates all the test data along with its respective results, which have been obtained through SVM and LSTM models. Analysing the table, it becomes evident that the overall performance under optimal and poor lighting conditions is not very different using our algorithm. It is partly because of the strong robustness of the MoveNet and its human gesture feature inputs for the machine learning algorithms.

It's also worth noting that when a sufficient volume of data is fed into the system - for instance, more than 1000 data samples - even traditional machine learning algorithms can yield results that are on par with, if not better than, those obtained from commonly used deep learning approaches. This suggests that while newer methods like deep learning are gaining popularity, traditional machine learning methods, given the right conditions and sufficient data, still hold value and can deliver robust results.

TABLE I: Classifiers' performance based on different testing conditions and algorithms

Data	Metric	SVM	LSTM 1 hidden layer	LSTM 2 hidden layers
General	Accuracy	0.634	0.649	0.655
	Precision	0.473	0.499	0.512
	Recall	0.366	0.433	0.358
	F1	0.413	0.464	0.422
No light	Accuracy	0.654	0.636	0.669
	Precision	0.500	0.464	0.534
	Recall	0.307	0.343	0.335
	F1	0.380	0.395	0.412
Light	Accuracy	0.618	0.615	0.599
	Precision	0.500	0.464	0.534
	Recall	0.314	0.289	0.186
	F1	0.348	0.327	0.231
Occlusion	Accuracy	0.635	0.697	0.705
	Precision	0.527	0.599	0.637
	Recall	0.0.463	0.638	0.537
	F1	0.493	0.618	0.583

However, F1 scores are generally low. This is due to the unbalanced dataset. Due to the nature of the fall events, the positive data (fall) are far less than non-fall. Using a weighted F1 score can clearly show that the average performance, even the best model, is less than 61.8%, which is still a comparable result with similar approaches such as [13].

One way to improve the F1 scores could be to capture and include more videos and frames of fall events in our dataset. Considering the tensor representation of pixel values and positional details of fall events, another approach could be the augmentation of the minority class (i.e., fall/positive events from the samples) to create more representation of falls, thereby addressing this imbalance.

To further compare the proposed methods against other methods. We also recorded the confusion matrix of the SVM model for fall events with above-mentioned four types of data as shown in the Fig. 4

By examining the relationship in our model's accuracy across the data group, a steady increase in accuracy is observed moving from "General" Data through to "Data without Light" from 0.634 to 0.654. However, a decrease in accuracy is experienced for "Data with Light" compared to other data groups, as shown from the plot of accuracy against the data group as shown above. This observation in performance is caused by the radial basis function(RBF) kernel in SVM to noise in data. The low bias of the RBF kernel to light noise in the data accounts for this decrease in accuracy performance for "Data with Light" as the light noise in the data creates overlapping of classes in the data, leading to decreased classification performance of the SVM when using RBF.

As shown in the Tab. II, following further experiments on different levels of illumination, we also conducted a group of experiments across the illumination groups for an insight into the model against illumination variations.

Examining the trend across the illumination level, we noticed relative changes in the accuracy of our model as the

TABLE II: Classifiers' performance based on different illumination conditions (125, 130 and 221 lux)

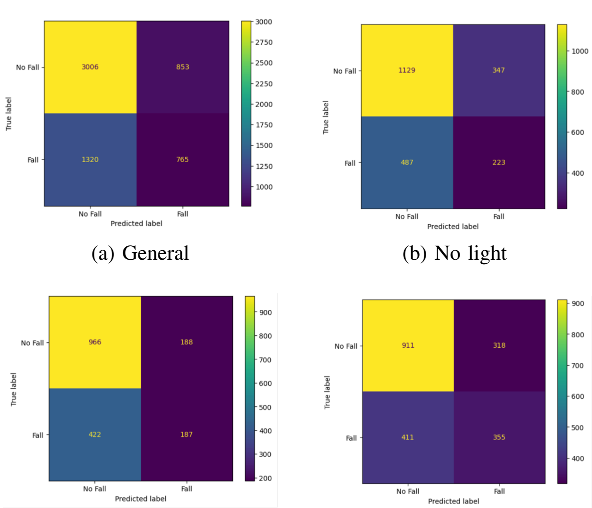
Illumination	Metric	SVM	LSTM 1 hidden layer	LSTM 2 hidden layers
125 lux	Accuracy	0.560	0.684	0.693
	Precision	0.932	0.930	0.960
	Recall	0.442	0.622	0.612
	F1	0.600	0.746	0.747
130 lux	Accuracy	0.557	0.706	0.598
	Precision	0.948	0.915	0.938
	Recall	0.379	0.633	0.447
	F1	0.541	0.748	0.605
221 lux	Accuracy	0.222	0.288	0.429
	Precision	0.355	0.447	0.745
	Recall	0.162	0.154	0.257
	F1	0.222	0.230	0.383

illumination level increased with peak performance achieved at the illumination of 130 lux and a decreasing accuracy in our model as the illumination level increased through 221 lux. The accuracy peaked at the illumination level of 130 lux with an accuracy of 0.706, with our model's accuracy decreasing to 0.418 as the illumination level increased further to 221 lux.

VI. Conclusion

This paper explored a new approach to fall detection by applying vision-based techniques and machine-learning algorithms to video data. We adopted the MoveNet model for human pose estimation due to its superior efficiency and accuracy. Our experiments involved using both classical machine learning algorithms, specifically SVM, and neural network models such as LSTM. Our primary focus was to examine the impact of varying lighting conditions on the performance of our fall detection system. The results indicate that the proposed system is robust and can effectively detect falls even under less than ideal lighting conditions.

In the future, we plan to further refine our model by incorporating more diverse and challenging real-world scenarios in the healthcare domain. This includes situations with varying environmental factors, such as cluttered spaces, multiple individuals, and more diverse and active movements. We also aim to explore the potential specific fall types in healthcare. Also, some more advanced models base on transformers and clips can also be tested and implemented in our system when running cost is reduced for real-time social robot systems.



(a) General (b) No light (c) Light (d) Occlusion

Fig. 4: Confusion Matrices from different data type using SVM classification

Acknowledgment

This research is funded by Sheffield Hallam Computing Research Fund for postgraduate students. For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY) license to any Accepted Manuscript version arising.

References

[1] W. H. O. Ageing and L. C. Unit, *WHO global report on falls prevention in older age*. World Health Organization, 2008.

[2] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[3] H.-S. Fang, J. Li, H. Tang, C. Xu, H. Zhu, Y. Xiu, Y.-L. Li, and C. Lu, "Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[4] R. Espinosa, H. Ponce, S. Gutiérrez, L. Martínez-Villaseñor, J. Brieva, and E. Moya-Albor, "A vision-based approach for fall detection using multiple cameras and convolutional neural networks: A case study using the up-fall detection dataset," *Computers in biology and medicine*, vol. 115, p. 103520, 2019.

[5] I. Charfi, J. Miteran, J. Dubois, M. Atri, and R. Tourki, "Definition and performance evaluation of a robust svm based fall detection solution," in *2012 eighth international conference on signal image technology and internet based systems*, pp. 218–224, IEEE, 2012.

[6] N. Zerrouki and A. Houacine, "Combined curvelets and hidden markov models for human fall detection," *Multimedia Tools and Applications*, vol. 77, pp. 6405–6424, 2018.

[7] N. Thome, S. Miguet, and S. Ambellouis, "A real-time, multiview fall detection system: A lhmm-based approach," *IEEE transactions on circuits and systems for video technology*, vol. 18, no. 11, pp. 1522–1532, 2008.

[8] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Robust video surveillance for fall detection based on human shape deformation," *IEEE Transactions on circuits and systems for video Technology*, vol. 21, no. 5, pp. 611–622, 2011.

[9] N. Lu, Y. Wu, L. Feng, and J. Song, "Deep learning for fall detection: Three-dimensional cnn combined with lstm on video kinematic data," *IEEE journal of biomedical and health informatics*, vol. 23, no. 1, pp. 314–323, 2018.

[10] Google, "Movenet: Ultra fast and accurate pose detection model," tech. rep., Google, 2024.

[11] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

[12] J. C. E. Guerrero, E. M. España, M. M. Añasco, and J. E. P. Lopera, "Dataset for human fall recognition in an uncontrolled environment," *Data in brief*, vol. 45, p. 108610, 2022.

[13] C.-B. Lin, Z. Dong, W.-K. Kuan, and Y.-F. Huang, "A framework for fall detection based on openpose skeleton and lstm/gru models," *Applied Sciences*, vol. 11, no. 1, p. 329, 2020.