

Real-Time Human Fall Detection using a Lightweight Pose Estimation Technique

Ekram Alam^{1,3}, Abu Sufian², Paramartha Dutta³, and Marco Leo⁴

¹ Department of Computer Science, Gour Mahavidyalaya, West Bengal, India

² Department of Computer Science, University of Gour Banga, India

³ Department of Computer and System Sciences, Visva-Bharati University, India

⁴ National Research Council of Italy, Institute of Applied Sciences and Intelligent Systems, 73100 Lecce, Italy

Abstract. The elderly population is increasing rapidly around the world. There are not enough caretakers for them. Use of AI-based in-home medical care systems is gaining momentum due to this. Human fall detection is one of the most important tasks of medical care system for the aged people. Human fall is a common problem among elderly people. Detection of a fall and providing medical help as early as possible is very important to reduce any further complexity. The chances of death and other medical complications can be reduced by detecting and providing medical help as early as possible after the fall. There are many state-of-the-art fall detection techniques available these days, but the majority of them need very high computing power. In this paper, we proposed a lightweight and fast human fall detection system using pose estimation. We used ‘Movenet’ for human joints key-points extraction. Our proposed method can work in real-time on any low-computing device with any basic camera. All computation can be processed locally, so there is no problem of privacy of the subject. We used two datasets ‘GMDCSA’ and ‘URFD’ for the experiment. We got the sensitivity value of 0.9375 and 0.9167 for the dataset ‘GMDCSA’ and ‘URFD’ respectively. The source code and the dataset GMDCSA of our work are available online to access.

Keywords: Fall Detection, Pose Estimation, GMDCSA, Movenet, Lightweight Fall Detection, Real-time Fall Detection

1 Introduction

Human fall is one of the major reasons for hospitalization in elder people around the world [1]. Detection of human falls is very vital so that medical help can be provided as early as possible. Human fall detection can be done using wearable, ambient, or vision sensors [2]. Vision-based fall detection system is more suitable, especially for elder people [3]. There is no need to attach the vision sensor to the body like wearable sensors. Wearable sensors need to be charged frequently whereas vision sensors can work on a direct home power supply. Human fall detection is one of the useful application of computer vision [4], [5]. In this paper, we have proposed a lightweight human fall detection system using pose

estimation [6], [7]. We have used a lightweight and fast pose estimation model ‘Movenet Thunder’ [8] for our work. The main contributions of this work are as given below.

Real Time ‘Movenet’ processes the video with 30+ FPS [9] (real-time) in the majority of current low computing devices like mobile phones, laptops, and desktops. So the proposed system can work in real-time on these devices. We tested our work on an average computing laptop with inbuilt webcam.

Lightweight The proposed system does not required very high computing power and can work on any normal laptop/desktop or mobile device.

Local Computation All computation can be processed locally. There is no personal data (images/frames) transfer from edge [10] to the cloud and vice versa. Only the output (fall) is sent to the caretaker center for necessary medical help. In this way, our system also preserves the privacy of the subject.

GMDCSA Dataset A new fall detection dataset named GMDCSA was introduced.

The rest of the paper is structured as follows. Section 3 describes related work briefly. Section 2 gives an overview of the pose estimation using ‘Movenet’. Section 4 discusses the methodology of our work. Section 5 describes the datasets which were used in this work. Section 6 provides the results of this work in the form of different metrics. Finally, section 7 concludes the proposed work with possible future scopes.

2 Background Study

We have used a lightweight pose estimation model named ‘Movenet Thunder’ [8]. This model accepts an RGB frame or image of the size 256 x 256 and extracts the normalized coordinate and confidence values of the 17 key-points of the human body joints. The 17 key-points are shown in Figure 1. The indices (from 0 to 16), Keypoints, Y values, X values, and confidences value of a sample image are shown in Table. 1. The values of y,x, and confidence are normalized from 0 to 1. The top left position is the origin(0,0) and the bottom right position has the value (1,1). When the keypoints are clearly visible then confidence tends to 1 (100%) otherwise it tends to 0 (0%).

3 Related Work

This section briefly describes some recent related works. Asif et al. [11] introduced a single-shot fall detection technique using 3D poseNet. Chen et al. [12] proposed a 3D posed estimator which was used as input for the fall detection network. Apicella and Snidaro [13] proposed a fall detection method based on CNN, RNN and PoseNet pose estimation. Leite et al. [14] introduced a multi (three) channel CNN-based fall detection system. Optical flow, pose estimation, and visual rhythm were used as inputs for three different streams of the CNN.

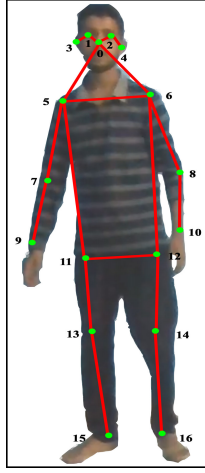


Fig. 1: 17 Keypoints of the Movenet pose estimation model

Table 1: Index, Keypoint, Y value, X value and confidence of an sample image

Index	Keypoint	Y Value	X Value	Confidence
0	Nose	0.22416662	0.579579	0.7201656
1	Left Eye	0.20926172	0.5974146	0.8043867
2	Right Eye	0.20485064	0.5642889	0.5905826
3	Left Ear	0.22323	0.6126661	0.7964257
4	Right Ear	0.21771489	0.5370738	0.7529471
5	Left Shoulder	0.3235461	0.6375601	0.8950565
6	Right Shoulder	0.2964768	0.48282918	0.65825576
7	Left Elbow	0.43468294	0.63684213	0.7667525
8	Right Elbow	0.42770475	0.4406372	0.8829603
9	Left Wrist	0.54110587	0.6462866	0.6282949
10	Right Wrist	0.5392799	0.42464092	0.8215329
11	Left Hip	0.54277164	0.57565194	0.85804665
12	Right Hip	0.53679305	0.48321638	0.88962007
13	Left Knee	0.69595444	0.609515	0.8796475
14	Right Knee	0.7019378	0.46842176	0.6786141
15	Left Ankle	0.85588527	0.56420994	0.7951814
16	Right Ankle	0.8588409	0.47616798	0.82729894

OpenPose [15] was used for pose estimation. Chen et al. [16] proposed a fall detection system using the Yolov5 network [17]. Liu et al. [18] proposed a fall detection system based on BlazePose-LSTM. This system was introduced especially for seafarers. Beddiar et al. [19] introduced a work based on the angle formed by the line from the centroid of the human face to the centroid of the hip to the line formed from the centroid of the hip to the horizontal axis. Am-saprabhaa et al. [20] proposed a multimodal gate feature-based fall detection system.

4 Methodology

The methodology of the proposed work is shown in Figure 2. The input can

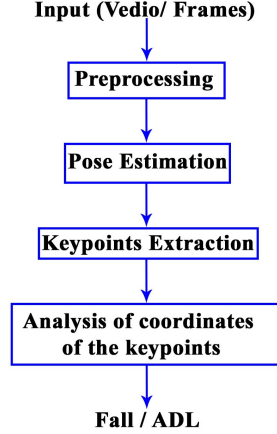


Fig. 2: Proposed work methodology

be an image, frames of a video, or the live video stream. The input images / frames were resized to 256 x 256 as preprocessing before feeding it to the Movenet. After preprocessing, pose estimation was done using the Movenet. The Movenet extracts the key-point co-ordinates with their confidence score as shown in Table 1. Confidence score can vary from 0 (0%) to 1 (100%) for each keypoints. If all key-points with very low confidence are also used for fall detection then it might select the wrong keypoints and this will reduce the performance of the system. If the high confidence value threshold are used it may ignore some good keypoints which might be useful for the detection of the fall. After experimenting with different threshold values of the confidence score, we finally selected 0.5 as threshold value because it gave good results. We have selected only those key-points whose confidence scores are greater than 0.5. The fall activity and sleeping activity are very similar and there are high chances of detecting a sleeping activity as fall. If there is a sleeping like activity on the floor then the system should detect it as fall activity, but if there is a sleeping like activity on the bed then the system should detect it as ADL (not fall) activity. To filter out this we compared the approximate y value of the top of the bed with the y value of the nose, eyes, ears, shoulders, elbows and wrists (upper body part). If the y value of these key-points (upper body part) are greater than the approximate y value of the top of the bed, then the activities of these frames are not fall and filtered out for the fall detection. After that the coordinates (x, y values) of the keypoints of the upper body parts were compared with the coordinates of key-points of the lower body parts(hips, knees, ankles). If the differences of the y value of the upper body keypoints (UBK) with lower body

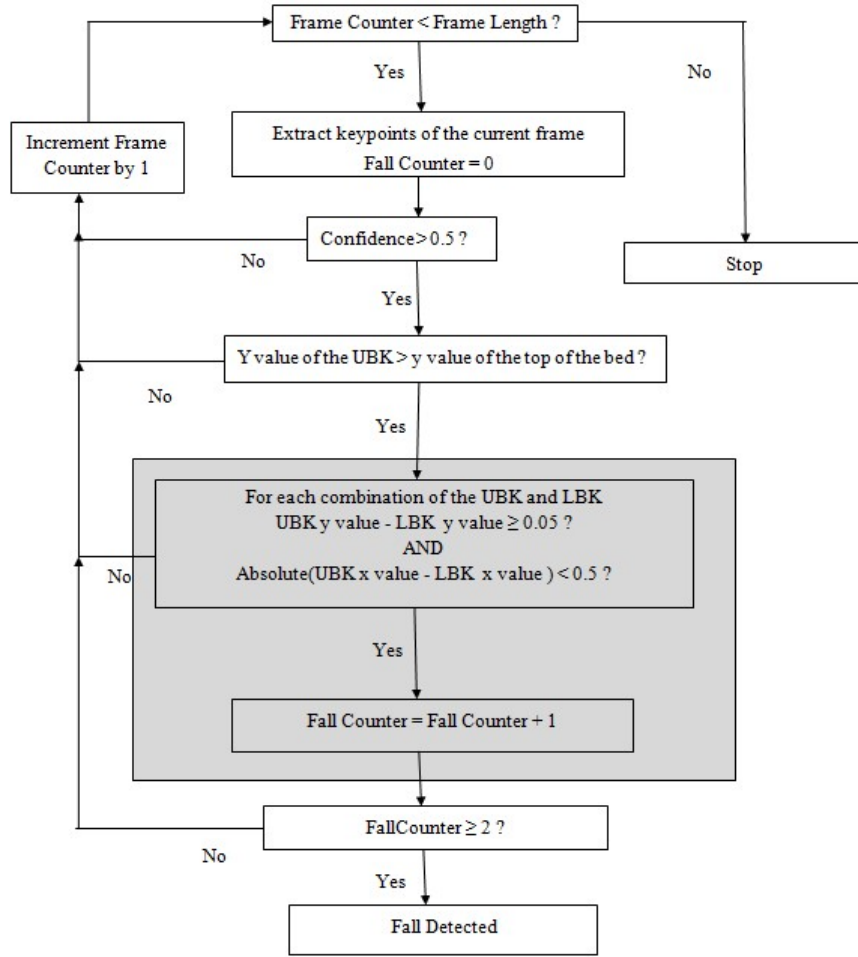


Fig. 3: Proposed work methodology flowchart

keypoints (LBK) is less than or equal to 0.05 (threshold-y) and the absolute differences of the x value is greater than 0.5 (threshold-x) then there is a chance of a Fall and the fall counter is increased by one. The selection of the values for the threshold-y and threshold-x were done after doing many experiments with different values. These values gave the best results. If in the next frame this is false then counter reset to 0. If this happens continuously for 2 or more frames (minimum counter value 2), then the system detects it as a fall, and a fall alert is sent. Detail of the analysis is shown in Figure 3. The source code of the proposed work is available here <https://github.com/ekramalam/RTHFD>.

5 Dataset

We used two datasets for the proposed experiment, the URFD [21] dataset and a dataset (GMDCSA) created by us. The URFD dataset contains 40 ADL (not fall) activities and 30 fall activities. The GMDCSA dataset contains 16 ADL (not fall) activities and 16 Fall activities. The GMDCSA dataset has been created by performing the fall and the ADL activities by a single subject wearing different set of clothes. The web camera of a laptop (HP 348 G5 Laptop : Core i5 8th Gen/8 GB/512 GB SSD/Windows 10) was used to capture the activities. The description of the ADL and Fall video sequences of the GMDCSA dataset are shown in Table 2 and 3 respectively. The link to access this dataset is as follows https://drive.google.com/drive/folders/1ohDEXki8Wz12cJ1XzyKIK4T6y1_hAf3p?usp=sharing.

Table 2: GMDCSA Dataset: ADL Activities

File Name	Length	Description
01.mp4	08 sec	Sitting on the bed to sleeping right side on the bed. Face towards camera
02.mp4	06 sec	Sitting on the bed to sleeping left side on the bed. Face towards camera.
03.mp4	06 sec	Sitting on the bed to sleeping left side on the bed. Face towards ceiling.
04.mp4	05 sec	From sleeping left side on the bed (Face towards ceiling) to sitting on the bed.
05.mp4	10 sec	Coming to the bed and reading book in sitting position. Front view of the subject. One leg folded.
06.mp4	12 sec	Sitting on the bed (front view) to reading the book while supporting towards the wall. Side view, Leg straight.
07.mp4	07 sec	Reading book while sitting on the chair. Front view.
08.mp4	06 sec	Walking in the room.
09.mp4	04 sec	Reading book while walking in the room.
10.mp4	03 sec	Reading book while walking in the room.
11.mp4	09 sec	Walking to sitting on the chair and then reading book.
12.mp4	07 sec	Reading book while sitting on the chair to stand up and keeping the book on the chair and going out of the room.
13.mp4	06 sec	Walking to sitting on the chair (side view).
14.mp4	05 sec	Sitting on the chair (side view) to walking.
15.mp4	07 sec	Walking to picking a mobile phone from the ground and then sitting on the chair.
16.mp4	03 sec	Picking the mobile phone from the ground while sitting on the chair

6 Result

The performance of a model can be measured using different evaluation metrics [1] like sensitivity, specificity, precision, etc. The values of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) are needed to calculate the values of these metrics. The value of TP, TN, FP, and FN can easily be found from the confusion matrix as shown in Figure 4. The value of TP, TN, FP, and FN for the two dataset GMDCSA and URFD are shown in Table 4. Table 5 shows the values of sensitivity, specificity, precision, false positive rate, false negative rate, accuracy, and F1 Score. These values can be calculated using the values of TP, TN, FP, and FN as shown in Table 4. Table 5 also shows the expressions of the corresponding metrics. The Sensitivity is more

Table 3: GMDCSA Dataset: Fall Activities

File Name	Length	Description
01.mp4	06 sec	Falling from sitting on the chair to the ground. Left side Fall. Full body not visible.
02.mp4	06 sec	Falling from sitting on the chair to the ground. Left side Fall. Full body not visible.
03.mp4	05 sec	Falling from sitting on the chair to the ground. Left side fall.
04.mp4	04 sec	Falling from sitting on the chair to the ground. Right side fall.
05.mp4	05 sec	Walking to falling. Right side fall.
06.mp4	05 sec	Walking to falling. Right side fall.
07.mp4	05 sec	Walking to falling. Right side fall.
08.mp4	04 sec	Walking to falling. Left side fall.
09.mp4	04 sec	Standing position to falling. Forward Fall. Full body (head) not visible.
10.mp4	04 sec	Standing position to falling. Forward Fall. Full body (right eye) not visible.
11.mp4	06 sec	Standing position to falling. Backward Fall.
12.mp4	06 sec	Standing position to falling. Backward Fall.
13.mp4	04 sec	Standing position to falling. Backward fall. Full body (head) is not visible.
14.mp4	05 sec	Standing position to falling. Backward fall. Full body (both ankles) is not visible.
15.mp4	06 sec	Sitting on the chair (side view) to right side fall.
16.mp4	06 sec	Sitting on the chair (side view) to left side fall.

	True Positive (Fall)	True Negative (ADL)		True Positive (Fall)	True Negative (ADL)
Predicted Positive (Fall)	15	1	Predicted Positive (Fall)	22	11
Predicted Negative (ADL)	1	15	Predicted Negative (ADL)	2	29

(a) GMDCSA

(b) URFD

Fig. 4: Confusion Matrix

Table 4: The value of TP, TN, FP, and FN for the dataset GMDCSA and URFD

Dataset	TP	TN	FP	FN
GMDCSA	15	15	1	1
URFD	22	29	11	2

Table 5: Results of the experiment

Metric	Expression	Dataset	
		GMDCSA	URFD
Sensitivity	$TP / (TP + FN)$	0.9375	0.9167
Specificity	$TN / (FP + TN)$	0.9375	0.7250
Precision	$TP / (TP + FP)$	0.9375	0.6667
False Positive Rate	$FP / (FP + TN)$	0.0625	0.2750
False Negative Rate	$FN / (FN + TP)$	0.0625	0.0833
Accuracy	$(TP + TN) / (P + N)$	0.9375	0.7969
F1 Score	$2TP / (2TP + FP + FN)$	0.9375	0.6216

important than other metrics for any medical classification problem like human fall detection. The values of sensitivity are 0.9375, and 0.9167 for GMDCSA and URFD respectively. These values are good enough for a lightweight system.

The specificity for GMDCSA is 0.9375 whereas for URFD it is 0.7250. The performance of our model is better using the GMDCSA dataset than the URFD dataset. This may be because the ADL activities of URFD contained many complex falls-like activities. Some of these activities were classified wrongly as falls by our system.



Fig. 5: Sample outputs using the GMDCSA dataset

Some of the sample outputs of this experiment using the GMDCSA and URFD datasets are shown in Figure 5 and 6 respectively. The captions of the subfigure tell whether the frames are from the ADL sequence or the fall sequence. The number in the brackets of the caption is the file name of the video of the corresponding dataset. Figure 5 shows three sample ADL frames and three fall frames from the GMDCSA dataset video sequences. Similarly, Figure 6 shows three sample ADL frames and three fall frames from the GMDCSA dataset video sequences.

7 Conclusion and Future Scope

In this paper, we proposed a lightweight and fast human fall detection system using ‘Movenet Thunder’ pose estimation. Our proposed system is very fast and requires very low computing power. It can run easily in real-time on any low-computing device like mobile, laptop, desktop, etc. All computation is done locally, so it also preserves the privacy of the subject. The metrics are also good enough considering the low computing requirement of the system. The proposed

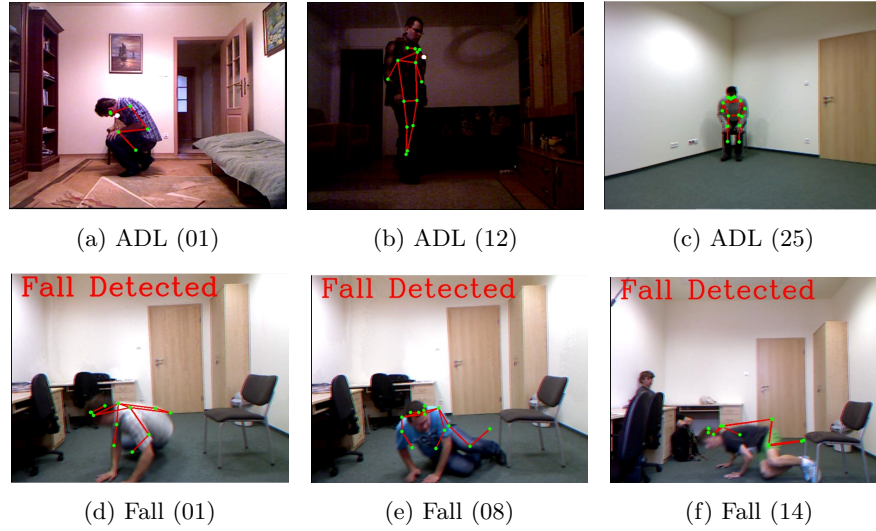


Fig. 6: Sample outputs using the URFD dataset

technique gave good results using the GMDCSA dataset. The sensitivity values are good for both datasets. The Movenet pose estimation model is a fast and lightweight model, but its accuracy is moderate. Also, our system can not work for more than one subject at the same time. In the future, we are thinking to improve our system so that it can work in multi-person [22] environments with high accuracy while maintaining the low computing requirement.

References

1. E. Alam, A. Sufian, P. Dutta, M. Leo, Vision-based human fall detection systems using deep learning: A review, *Computers in Biology and Medicine* (2022) 105626.
2. Z. Wang, V. Ramamoorthy, U. Gal, A. Guez, Possible life saver: A review on human fall detection technology, *Robotics* 9 (3) (2020) 55.
3. J. Gutiérrez, V. Rodríguez, S. Martín, Comprehensive review of vision-based fall detection systems, *Sensors* 21 (3) (2021) 947.
4. E. Alam, A. Sufian, A. K. Das, A. Bhattacharya, M. F. Ali, M. H. Rahman, Leveraging deep learning for computer vision: A review, in: *2021 22nd International Arab Conference on Information Technology (ACIT)*, IEEE, 2021, pp. 1–8.
5. X. Wang, J. Ellul, G. Azzopardi, Elderly fall detection systems: A literature survey, *Frontiers in Robotics and AI* 7 (2020) 71.
6. T. L. Muneá, Y. Z. Jembre, H. T. Weldegebriel, L. Chen, C. Huang, C. Yang, The progress of human pose estimation: a survey and taxonomy of models applied in 2d human pose estimation, *IEEE Access* 8 (2020) 133330–133348.
7. Y. Chen, Y. Tian, M. He, Monocular human pose estimation: A survey of deep learning-based methods, *Computer Vision and Image Understanding* 192 (2020) 102897.

8. R. Bajpai, D. Joshi, Movenet: A deep neural network for joint profile prediction across variable walking speeds and slopes, *IEEE Transactions on Instrumentation and Measurement* 70 (2021) 1–11.
9. MoveNet: Ultra fast and accurate pose detection model. — TensorFlow Hub — tensorflow.org, <https://www.tensorflow.org/hub/tutorials/movenet>, [Accessed 21-Oct-2022].
10. A. Sufian, E. Alam, A. Ghosh, F. Sultana, D. De, M. Dong, Deep learning in computer vision through mobile edge computing for iot, in: *Mobile Edge Computing*, Springer, 2021, pp. 443–471.
11. U. Asif, S. Von Cavallar, J. Tang, S. Harrer, Sshfd: Single shot human fall detection with occluded joints resilience, *arXiv preprint arXiv:2004.00797* (2020).
12. Z. Chen, Y. Wang, W. Yang, Video based fall detection using human poses, in: *CCF Conference on Big Data*, Springer, 2022, pp. 283–296.
13. A. Apicella, L. Snidaro, Deep neural networks for real-time remote fall detection, in: *International Conference on Pattern Recognition*, Springer, 2021, pp. 188–201.
14. G. V. Leite, G. P. da Silva, H. Pedrini, Three-stream convolutional neural network for human fall detection, in: *Deep Learning Applications*, Volume 2, Springer, 2021, pp. 49–80.
15. Z. Cao, T. Simon, S.-E. Wei, Y. Sheikh, Realtime multi-person 2d pose estimation using part affinity fields, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7291–7299.
16. T. Chen, Z. Ding, B. Li, Elderly fall detection based on improved yolov5s network, *IEEE Access* 10 (2022) 91273–91282.
17. Ultralytics, Yolov5, <https://github.com/ultralytics/yolov5>, [Accessed 14-Jan-2023].
18. W. Liu, X. Liu, Y. Hu, J. Shi, X. Chen, J. Zhao, S. Wang, Q. Hu, Fall detection for shipboard seafarers based on optimized blazepose and lstm, *Sensors* 22 (14) (2022) 5449.
19. D. R. Beddiar, M. Oussalah, B. Nini, Fall detection using body geometry and human pose estimation in video sequences, *Journal of Visual Communication and Image Representation* 82 (2022) 103407.
20. M. Amsaprabhaa, et al., Multimodal spatiotemporal skeletal kinematic gait feature fusion for vision-based fall detection, *Expert Systems with Applications* 212 (2023) 118681.
21. B. Kwolek, M. Kepski, Human fall detection on embedded platform using depth maps and wireless accelerometer, *Computer methods and programs in biomedicine* 117 (3) (2014) 489–501.
22. M. Kocabas, S. Karagoz, E. Akbas, Multiposenet: Fast multi-person pose estimation using pose residual network, in: *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 417–433.