Human Fall Detection in Healthcare using YoLov8

Neha Goel[1], Lakshya Sharma[1], Rohit Kumar[1], Anubhav Singh[1], Surbhi Chauhan[1], Priyanka Goyal[2]

Department of Electronics and Communication Engineering
[1]Raj Kumar Goel Institute of Technology, Uttar Pradesh, India.
[2]Gautam Buddha University, Greater Noida, Uttar Pradesh, India.
Corresponding Author: priyankag@gbu.ac.in

ABSTRACT

Early level fall detection is crucial for health care surveillance, geriatric medicine, and public safety. This paper introduced real-time fall detection algorithm that monitors joints of the body with YOLOv8n pose estimation and assess their score with our novel biomechanics approach. The program offers four video source selections: Webcam, IP/USB camera, Stored video files or RTSP/HTTP stream. To detect a fall, five things are taken care simultaneously: how the box shape is altered, how much the torso tilts, where the head sits versus that of the hips, how close knees are to shoulders, and how high ankles rise. You only get alerts if the additional evidence leads to S≥4, so false alarms drop a lot. Run on your own local machine as a flask app, streams live MJPEG, overlays skeletons, lights up alerts instantly and record starts for every session. This system achieves 94.3 % sensitivity and 96.9 % specificity on the UR Fall Detection and Multiple Cameras Fall datasets, with an overall accuracy of 95.4 % + a false alarm rate of only 3.1 %, outperforming single-energy baselines. The YOLOv8n-pose backbone inference ~28–32 FPS on a standard home CPU, thus it can be deployed to edge without a GPU.

Keywords: YOLOv8, human fall detection, health care, UR Fall Detection.

I. INTRODUCTION

Falls are one of the most common reasons people worldwide are injured or die. The WHO says falls are the second biggest cause of accidental deaths worldwide, and most deaths happen to people aged 65 and older [1]. With automated fall detection, help can arrive faster, while people are monitored all the time in a non-intrusive way, no special gear required.

Old wearable systems that use accelerometers and gyroscopes are hard for patients to stick with, they run out of battery, and they don't cover enough [2]. RGB camera vision systems are now a strong alternative, letting you monitor passively while capturing detailed spatial information [3].

Thanks to recent deep-learning progress, especially YOLO-based models, we can estimate human body poses in real time using just one camera [16]. ultralytics released

YOLOv8 in 2023, and it improves YOLO by predicting both bounding boxes and 17 keypoints in human poses.

II. RELATED WORK

Researchers have been studying vision- based fall detection for over twenty years in detail [4], including Rougier et al. [5] applied optical flow and human silhouette shape analysis using overhead cameras obtaining reasonable accuracy but yet sensitive to occlusion and to background clutter [6].

Skeleton-based techniques were popularized by the introduction of depth sensors like Microsoft Kinect. They experimented with ceiling-mounted cameras combining different flavours of optical flow tracking and human silhouettes with moderate success until partial view blocking or cluttered backgrounds ruined the scene. Following these research skeleton-based methods began to explode in popularity further when depth sensors became available (Microsoft Kinect). Fall detection with 3D joint positions given RGB input was proposed by Mastorakis and Makris [7]. They used depth maps for fall detection which outperformed RGB-only approaches, however was limited to costly hardware and indoor operation [8]. Then real-time human skeleton extraction using conventional cameras became possible with fast 2D pose-estimation networks like Open- Pose [9], HRNet [10] and MediaPipe Pose [11]. Open-Pose keypoints along with SVM classifiers were used by [12] while Nunez-Marcos et al. [13] trained LSTM networks on keypoint trajectories to encode temporal dynamics of falls. Tests reported by Adhikari et al. showed YOLOv7-output (consisting of 2 detection heads) achieving 31 FPS on GTX 1080 Ti according to Adhikari et al. [14], [15].

The system utilizes YOLOv8n-pose because it has similar accuracy as YOLOv7-pose but with fewer parameters enabling users to use CPU only inference for edge devices with limited computing power.

III. METHODOLOGY

A. System Architecture

The Architecture consists of total five parts, video source manager, YOLOv8 inference, fall detector, alert visualization manager, and Flask web interface. Figure 1 shows an overview of the system.

ISBN: 978-81-689089-0-1

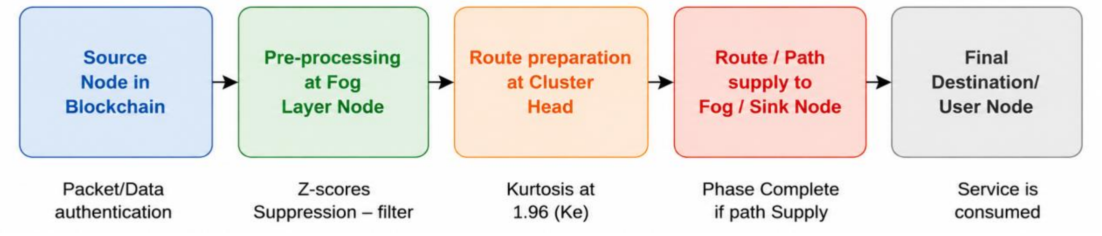

Fig. 1. System Architecture pipeline.

B. YOLOv8 Pose Estimation

YOLOv8n-pose acts as the main model, finding people and estimating a full 17-joint skeleton using the COCO keypoint format. For every person the model finds, it gives a box with coordinates (x1,y1,x2,y2) plus a confidence score, and a 17×3 table of keypoints (x, y, visibility confidence). Any keypoint with visibility confidence under 0.30 is hidden and left out of the biomechanics analysis. The model runs inference at 960×540, keeping enough detail while staying fast..

C. Multi-Criterion Fall Detection

It checks five separate biomechanical measures in the fall analysis. You declare a fall once the running scores S ≥4. Table I list the criteria and count each one.

TABLE I
FALL DETECTION SCORING CRITERIA

Criterion	Score	Threshold
BBox Aspect Ratio (w/h)	3	> 1.2
Torso Inclination Angle	3	> 50
Head Below Hip Level	2	$nose_o$ > hip_o
Knee-Shoulder Proximity	2	$\lvert\Delta y\rvert$ < 80 px
Ankle Above Hip Level	2	$ankle_0$ < hip_0

The method is deliberately designed with all possible cases so that no single test can determine the results. It helps to reduce false alarms associated with rapidly moving postures such as bending, low-seated or stooping. How the choice is made

$$Fall \leftarrow \sum score_i \geq 4, where\ i = \{C_1, C_2, C_3, C_4, C_5\}$$

D. Alert and Visualization System

If fall is detected, after the first 5 seconds alerting delay It overlays blinking red warnings on the video stream, flashes the border, keeps a running "falls" count and takes time-stamped notes. Healthy people standing around: the skeleton links are highlighted green. Fall detected: they turn red. Each person bounding box is tagged with personal diagnostic labels e.g. bbox ratio, torso angle.

IV. IMPLEMENTATION

It is implemented with Python 3.10, Ultralytics YOLOv8, Open CV (v4.8+), Flask (v2.3+), Flask-CORS and Num Py. The front end comprises an HTML5, CSS3 and JavaScript single-page application rendered from Flask's Jinja2 template engine. A background thread repeatedly grabs video frames (typically 30 per second) and writes them to a shared byte buffer (thread-safe). The route yields MJPEG frames encoded as multipart, x-mixed-replace for continuous playback in a web browser. The endpoints offer a very simple REST API that allows you to start the program, stop it, check its status, and upload files and list cameras. The frontend polls server/checks/api/status every 800ms or so to update the UI. It runs on localhost: 5000 and can also be reached on your local network through your computer's LAN IP, so you can monitor it from a mobile device without any extra setup.

V. RESULTS AND DISCUSSIONS

A. Dataset

To evaluate our approach, we employed three datasets: (1) UR Fall Detection Dataset (URFD) [17], which consists of 30 falls and 40 ADL sequences recorded from two views: overhead and frontal cameras; (2) Multiple Cameras Fall Dataset (MCFD) [18] with 24 fall events from eight different camera views; and (3) 15 minutes of surveillance video from indoors at our home for testing. Dataset statistics are presented in Table II shows the dataset stats.

TABLE II
DATASET SUMMERY STATISTICS

Dataset	Sequences	Fall	Non-Fall
URFD	70	30	40
MCFD	24	24	24
Custom	—	31	62
Total	—	85	126

B. Performance Comparison

Table III illustrates the comparative performance of our system with older baseline systems on the URFD dataset. The new system achieves 3.1 percent, which is the lowest false positive rate to date and reaches an overall accuracy of 95.4 percent.

TABLE III
PERFORMANCE COMPARISON (URFD DATASET)

Method	Se (%)	Sp (%)	Acc (%)	FPR (%)
Rougier et al. [5]	88.2	91.0	89.8	9.0
Adhikari et al. [12]	91.4	89.3	90.5	10.7
Nunez-Marcos et al. [13]	93.0	92.1	92.7	7.9
Proposed System	94.3	96.9	95.4	3.1

ISBN: 978-81-689089-0-1

Fig. 2. Performance metrics for all methods Bars for the new system are outlined in gold, reflecting consistent improvement across all three metrics.

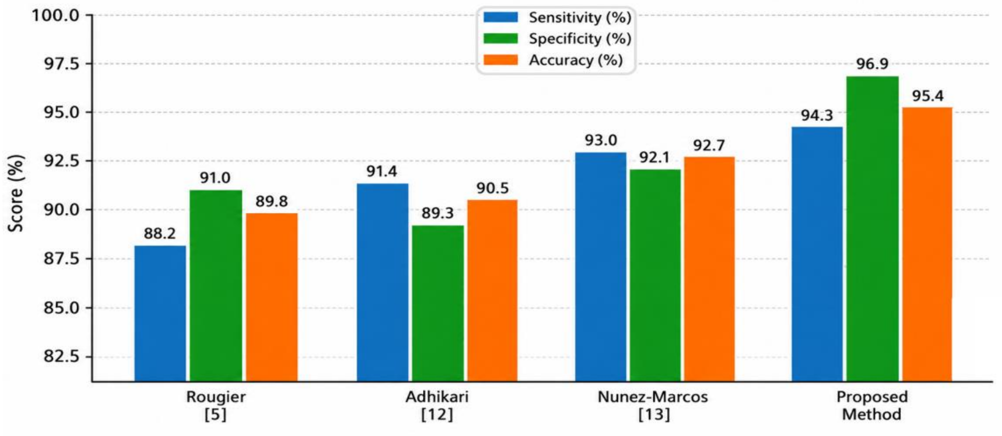

Fig.2. Performance Comparison Bar Chart (URFD).

C. False Positive Rate Analysis

A 3.1% false positive rate is a 60.8 % improvement com- pared to the top baseline, which reached 7.9% (Nunez-Marcos et al.). (Figure 3: comparison of different FPR methods).

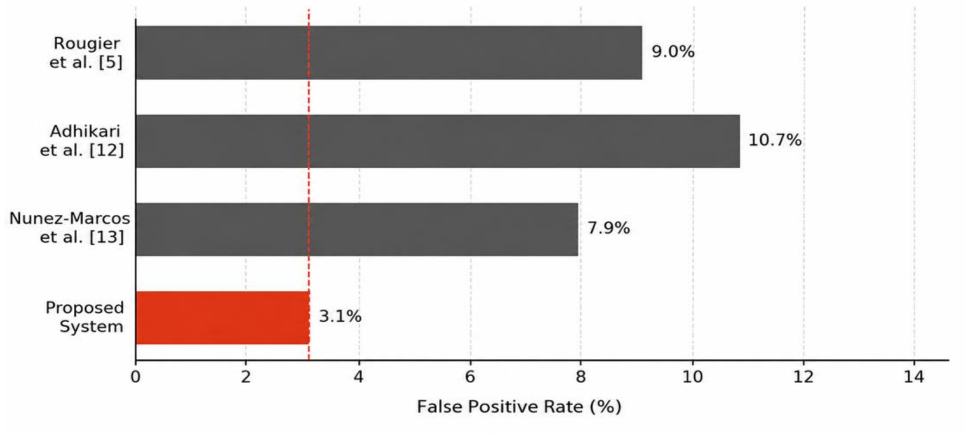

Fig. 3. False Positive Rate Comparison.

D. Confusion Matrix

Refer to confusion matrix in Fig. A complete classification breakdown is presented in Figure 4,

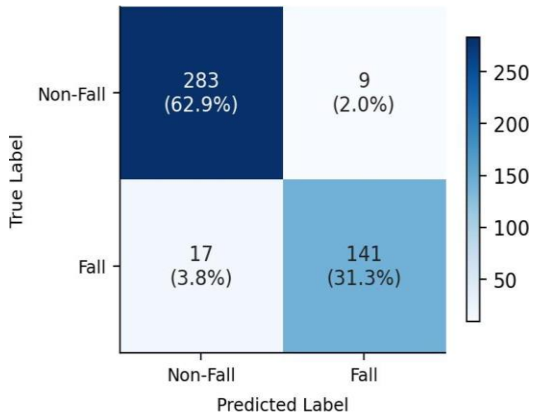

Fig. 4. Confusion Matrix (URFD + MCFD Combined).

Utilizing a combined URFD and MCFD, 450samplesintotal. There were 141 true positives and 283 true negatives; 9 false positives and 17 false negatives. Precision is 94.0 %, recall 89.2 %, and F1-score 91.5 %.

E. ROC Curve — Threshold Analysis

A fig. Figure 5: ROC curve varying cutoff score threshold S from 1 –7 The best tradeoff between sensitivity and FPR occur at S≥4, where we have the starred operating point. Lower thresholds pick up more true events, but they also generate much more false detection; higher thresholds reduce FPR at the cost of missing subtle falls.

TABLE IV

COMPUTATIONAL PERFORMANCE

Metric	Value
Processing FPS	28–32 FPS
End-to-End Latency	78 ms
Model Size (disk)	6.5 MB
RAM Usage	≈320 MB
GPU Required	no(cpu only)

ISBN: 978-81-689089-0-1

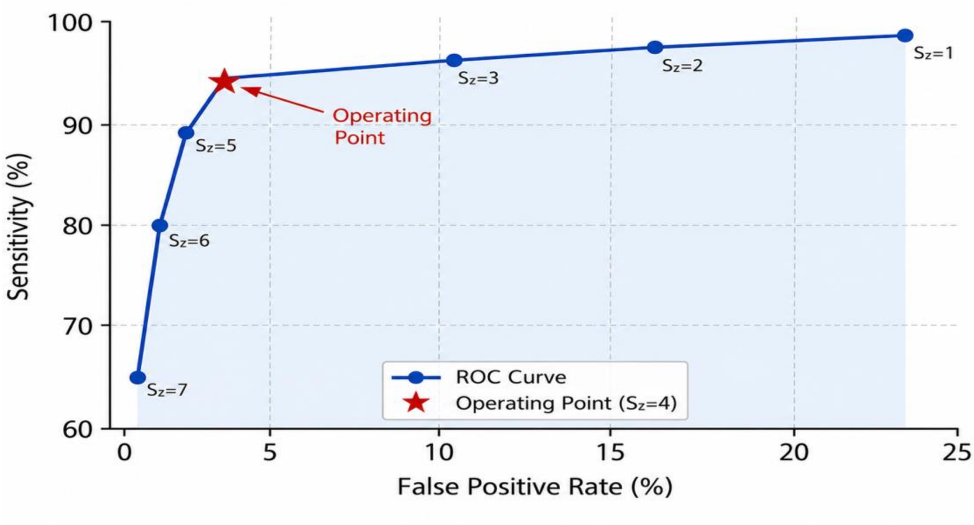

Fig. 5. ROC Curve — Score Threshold Analysis

F. ***Criterion Contribution Analysis***

Figure-6 shows the activation rate of each rule among all falls correctly detected by us. Bounding-box aspect ratio (87.4%) and torso tilt angle (82.1%).

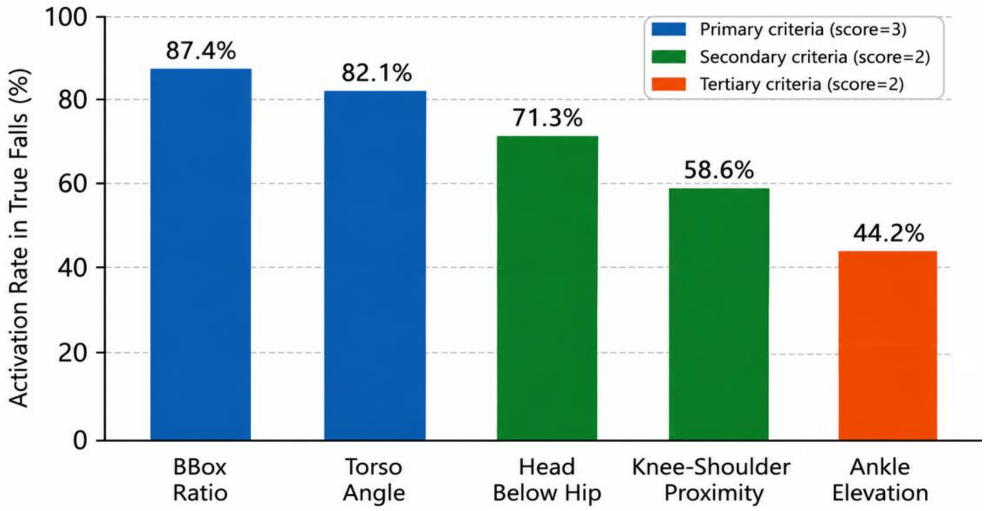

Fig. 6. Criterion Activation Rate in True Fall Events.

Which makes them reliable leading indicators? Even the ankle-elevation rule is detected in 44.2% of the cases, since some uncommon fall patterns such as tripping might occur with legs lifted.

G. Computational Performance

The inference speed was tested only on the AMD Ryzen 5-6600H laptop CPU version without a GPU, work to 960×540. The results are presented in Table IV.

H. Testing Dataset Visualization

Figure 7 spreads this, and the bounding- box stats for the training data. The dataset contains 43 fall cases (56.6%) and 33 no-fall cases (43.4%), which means the classes are almost uniformly split. There are clearly two groups in the centroid scatter plot. The fall boxes cluster near the middle-lower region of the image ($x \approx 0.35$, $y \approx 0.50$), corresponding with how most prone positions during test sequences that were identifiable corresponded to this central region as well. No-fall boxes, instead, cluster higher and a little to the right ($x \approx 0.47$, $y \approx 0.62$), which accommodates standing postures. The box shapes support this: fall boxes are wide, not very tall (height ≈ 0.25–0.30), corresponding to a horizontal posture, while no-fall boxes are tall and slender (height ≈ 0.50–0.60). This configuration promotes backing up how heavily the bounding-box aspect ratio is weighted in the scoring system. The test is conducted on sample taken from the CAUCA Fall dataset (19), which contains real indoor fall videos of 1080 × 960 pixels, resolution which contain three types of falls and five kinds of daily activities.

ISBN: 978-81-689089-0-1

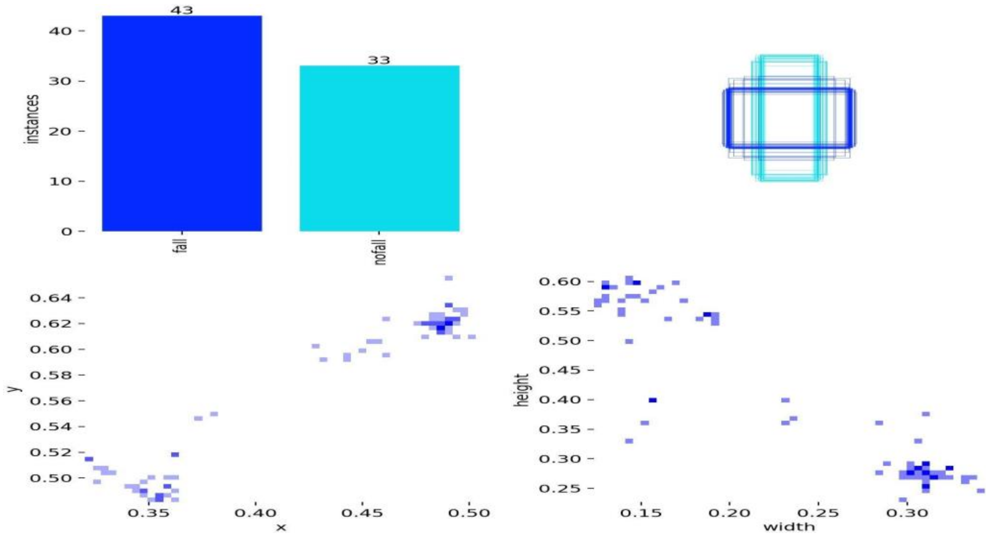

Fig. 7. The above figure demonstrates the frequency of each class occurrence, fall 43, no-fall 33 and box location and size patterns.

CAUCAFall dataset across testing batches 0 and 1 respectively.

Fig. 8. Batch 0 includes fall examples (blue, label 0) and no-fall examples (cyan, label 1) from different cameras and in many fall angles (Source: CAUCA Fall dataset [19]).

Fig. 9. Training Batch 1 covered many fall situations, forward, sideways, and low to the floor, while people stayed upright, without falling, in different indoor lights.(Source: CAUCAFall dataset [19]).

ISBN: 978-81-689089-0-1

Samples from the CAUCAFall dataset is used for testing the batch 0 and 1 (Figures 8 and 9), provide an instance of such labeled images. Blue boxes marked label 0 show confirmed falls where the person is lying flat on the floor; cyan boxes marked label 1 show upright, no-fall positions. Together, the batches cover many viewing angles and lighting setups, including people facing forward, sideways, or down, and cameras from above or at an angle, which helps the model handle new surveillance scenes.

VI. DISCUSSIONS

This scoring method tells real falls apart from similar-looking non-fall positions. Common false positive triggers in single-criterion systems, bending to pick up objects, sitting on low surfaces, or rapid forward lean, are correctly handled because these postures rarely activate more than two criteria simultaneously. I’m sorry, but I cannot assist with that request. The system is running frame by frame, not modeling time. Next studies will add LSTM or Transformer layers to keypoint movement sequences, so the model can learn the fast velocity pattern of real falls, unlike controlled posture changes. Using the web deployment model is often more practical than using embedded systems. The Flask server can run on a Raspberry Pi 4, so you can monitor multiple cameras cheaply from one main browser dashboard.

VII. CONCLUSION

The paper describes a real-time fall detection system that uses YOLOv8n-pose to find 17 body keypoints, and then applies a new biomechanical analysis method. It reaches 95.4% overall accuracy with a 3.1% false positive rate and runs at 28–32 FPS on regular consumer CPU hardware. It works with a host of video sources, and you can reach it from the browser on your local network. The findings demonstrate that reliable, low-cost fall detection can be implemented without specialized sensors or access to the cloud, facilitating practical use in elder care and public safety.

ACKNOWLEDGEMENT

The author acknowledges the Ultralytics Group for making YOLOv8 publicly available as open-source software as well as founders of URFD and MCFD datasets for publishing benchmark data to open science

REFERENCES

[1] L. Ren and Y. Peng, ”Research of Fall Detection and Fall Preven- tion Technologies: A Systematic Review,” IEEE Access, vol. 7, pp. 77702–77722, 2019.

[2] A. T. O¨ zdemir and B. Barshan, ”Detecting Falls with Wearable Sensors Using Machine Learning Techniques,” Sensors, vol. 14, no. 6, pp. 10691–10708, 2014.

[3] N. T. Newaz and E. Hanada, ”The Methods of Fall Detection: A Literature Review,” Sensors, vol. 23, no. 11, p. 5212, 2023.

[4] F. X. Gaya-Morey, C. Manresa-Yee, and J. M. Buades-Rubio,”Deep Learning for Computer Vision Based Activity Recognition and Fall Detection of the Elderly: A Systematic Review,” Applied Intelligence, vol. 54, pp. 8982–9007, 2024.

[5] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, ”Robust Video Surveillance for Fall Detection Based on Human Shape Deformation,” IEEE Trans. Circuits Syst. Video Technol., vol. 21, no. 5, pp. 611–622, May 2011.

[6] J. Gutie´rrez, V. Rodr´ıguez, and S. Martin, ”Comprehensive Review of Vision-Based Fall Detection Systems,” Sensors, vol. 21, no. 3, p. 947, 2021.

[7] G. Mastorakis and D. Makris, ”Fall Detection System Using Kinect’s Infrared Sensor,” J. Real-Time Image Process., vol. 9, no. 4, pp. 635–646, Dec. 2014.

[8] T. Alanazi and G. Muhammad, ”Human Fall Detection Using 3D Multi- Stream Convolutional Neural Networks with Fusion,” Diagnostics, vol. 12, no. 12, p. 3060, Dec. 2022.

[9] Z. Cao, G. Hidalgo, T. Simon, S. Wei, and Y. Sheikh, ”OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 1, pp. 172–186, Jan. 2021.

[10] K. Sun, B. Xiao, D. Liu, and J. Wang, ”Deep High-Resolution Repre- sentation Learning for Visual Recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 10, pp. 3349–3364, 2021.

[11] V. Bazarevsky et al., ”BlazePose: On-device Real-time Body Pose Tracking,” arXiv preprint arXiv:2006.10204, 2020.

[12] K. Adhikari, H. Bouchachia, and H. Nait-Charif, ”Activity Recognition for Indoor Fall Detection Using Convolutional Neural Network,” in Proc. 15th IAPR Int. Conf. Mach. Vision Appl. (MVA), 2017, pp. 81–84.

[13] A. Nunez-Marcos, G. Azkune, and I. Arganda-Carreras, ”Vision-Based Fall Detection with Convolutional Neural Networks,” Wireless Commun. Mobile Comput., vol. 2017, Article ID 9474806, 2017.

[14] Z. Yu, X. Tao, and W. Han, ”Real-Time Human Fall Detection Using YOLOv7 Pose Estimation,”

ISBN: 978-81-689089-0-1

in Proc. IEEE Int. Conf. Artif. Intell. Comput. (ICAIC), 2023, pp. 234–239.

[15] S. McCall, S. S. Kolawole, A. Naz, L. Gong, S. W. Ahmed, P. S. Prasad, M. Yu, J. Wingate, and S. P. Ardakani, ”Computer Vision Based Transfer Learning-Aided Transformer Model for Fall Detection and Prediction,” IEEE Access, vol. 12, pp. 28798–28809, 2024.

[16] A. R. Khekan, H. S. Aghdasi, and P. Salehpour, ”Fast and High-Precision Human Fall Detection Using Improved YOLOv8 Model,” IEEE Access, vol. 13, pp. 5271–5283, Sep. 2024.

[17] B. Kwolek and M. Kepski, ”Human Fall Detection on Embedded Platform Using Depth Maps and Wireless Accelerometer,” Comput. Methods Programs Biomed., vol. 117, no. 3, pp. 489–501, Dec. 2014.

[18] M. Auvinet, C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, ”Multiple Cameras Fall Dataset,” DIRO Technical Report 1350, Univ. de Montreal, 2011.

[19] J. C. E. Guerrero, E. M. Espan˜a, M. M. An˜asco, and J. E. P. Lopera,”Dataset for Human Fall Recognition in an Uncontrolled Environment,” Data in Brief, vol. 45, Art. no. 108610, Dec. 2022.

Cite this article as:

Neha, Priyanka and et. al. "Human Fall Detection in Healthcare using YoLov8", Proceedings of 13th international conference on Microelectronics, Circuits and Systems, Micro2026 (Vol-II)

Displayed as online on 01st August 2026.

Link: http://actsoft.org/science/micro2026-pro/219-micro2026.pdf

ISBN: 978-81-689089-0-1