A YOLOv5-Driven Edge Computing Framework for Real-Time Behaviour and Health Monitoring of Elderly Adults

J. Pradeep, P. Poojitha, P. Suguna, D. Umme Iman
Department of Electronics and Communication Engineering,
Aditya College of Engineering, Madanapalle, A.P., India.
E-mail: j.pradeepsn@gmail.com

Abstract:* Caring for older adults who live alone has long been difficult for families and health services, and that difficulty has grown as the share of seniors in the global population climbs steadily upward. This paper describes a monitoring platform we built and tested on a Raspberry Pi 4 single-board computer that tackles this challenge through a combination of on-device machine vision and pulse sensing. A USB camera streams live frames to a YOLOv5 object-detection model that we trained to classify standing, sitting and lying postures; a bounding-box aspect-ratio rule confirms that a posture change is genuine by requiring it to persist for at least two consecutive seconds and to place the person's bounding box below a room-specific floor calibration line. Running in parallel, a second software thread reads an analog photoplethysmography sensor through an MCP3008 ten-bit ADC connected over the SPI bus, estimates beats-per-minute using a sliding-window peak detector, and marks any reading outside 60–100 BPM as physiologically abnormal. Whenever either channel raises an alarm the system sounds a GPIO-connected buzzer and dispatches a structured SMTP e-mail to the registered caregiver. Because all detection and notification steps execute on the Raspberry Pi itself, raw video never crosses a network boundary, which directly addresses the privacy objection that most families raise when a camera is proposed for a home environment. Bench testing across thirty-seven scripted scenarios confirmed reliable identification of falls, self-harm gestures and abnormal movement, consistent heart-rate tracking under both resting and active conditions, and end-to-end alert latency below one second in every trial.

***Keywords:* behavior recognition, edge artificial intelligence, elderly care, fall detection, heart-rate monitoring, Internet of Things, Raspberry Pi, YOLOv5.**

I. Introduction

Across almost every country with reliable census records, the fraction of people aged sixty-five and above has risen without interruption for at least three decades. United Nations demographers project that this group, which stood at roughly nine per cent of the world population in 2020, will approach seventeen per cent by 2050. What makes this shift particularly relevant from an engineering standpoint is not the absolute count but the living arrangement: a growing proportion of these older adults spend most of their time at home, unsupervised, because either their families live elsewhere or in-home care is beyond their financial means.

Older adults living alone face a specific and well-documented set of hazards. Declining muscle strength, slower reflexes, reduced peripheral vision and underlying cardiovascular conditions combine to make routine activities genuinely risky. Falls are the most studied of these hazards. Longitudinal surveys consistently report that approximately one in three adults over sixty-five falls at least once per year, and clinical evidence shows that the duration of time spent on the floor before help arrives is among the most reliable predictors of long-term outcome. Equally concerning are silent cardiovascular changes—a gradual drift in resting heart rate or an intermittent arrhythmia—that often precede a serious cardiac event but produce no symptoms a non-specialist would recognize.

Human supervision, whether by a co-resident family member or a community nurse, is the conventional response to these risks. This arrangement has obvious limitations: it is expensive, it does not scale, and it leaves gaps whenever the carer is unavailable. A technology-based solution capable of continuous unattended observation, rapid alert generation and automatic caregiver notification would therefore represent a meaningful practical advance, provided it can be built at a cost that makes deployment in ordinary homes realistic.

Two independent lines of technical progress have recently converged to make this possible. Single-board computers, most notably the Raspberry Pi 4, now deliver enough processing capability to run a modern neural network in real time at a retail price well under one hundred US dollars. Concurrently, the YOLO family of object detectors has matured into a set of models that cover a wide range of hardware budgets. Unlike earlier detection pipelines that generated candidate regions and then classified them in a separate step, YOLO formulates detection as a single regression across the whole image, producing bounding-box coordinates and class probabilities in one forward pass and making it substantially faster for a given accuracy target. YOLOv5, the fifth generation, ships in four size variants from nano to extra-large; on a Raspberry Pi 4 the nano and small variants process several frames per second, which is sufficient for posture-level monitoring of a room.

We assembled a prototype that fuses two sensor channels on a single Raspberry Pi node: a USB camera whose output feeds a YOLOv5 detector, and a photoplethysmography pulse sensor whose analog voltage is digitized by an MCP3008

ADC over SPI. A compact rule-based layer evaluates both channels and decides whether an emergency alert is warranted. All computation happens on the device; the only data that ever leaves it are short structured e-mail notifications. The paper is organized as follows. Section II reviews prior work and identifies the gaps our design addresses. Section III documents the hardware architecture. Section IV describes the algorithm and alert logic. Section V presents bench-test results. Section VI draws conclusions and outlines directions for future work.

II. LITERATURE SURVEY

A. Wearable and Sensor-Based Monitoring

The earliest assistive devices aimed at elderly safety used accelerometers or gyroscopes housed in a pendant or wristband. When the sensor detected an acceleration spike consistent with a fall, it triggered a call to a monitoring center. Academic studies of these Personal Emergency Response Systems have reported detection accuracies in the eighty to ninety per cent range under controlled conditions, and some commercial products have been available for decades. The persistent weakness of this category, however, is user compliance. Longitudinal studies that followed real users over six months found adherence rates well below fifty per cent: older adults remove the device to bathe, find the strap uncomfortable or simply forget to recharge the battery. A device not being worn cannot detect a fall, and this fundamental limitation has driven researchers toward camera-based alternatives that require nothing of the resident.

B. Threshold-Based Camera Systems

Fixed cameras combined with background subtraction or frame-differencing algorithms eliminate the compliance problem entirely. The resident needs to do nothing except be in the room. The trade-off is a severe loss of semantic understanding. Background-subtraction methods cannot distinguish a genuine fall from a slow controlled lie-down, and they generate false alarms whenever a curtain moves, a pet walks through frame or outdoor lighting changes abruptly. Most published prototypes in this category were evaluated in laboratory settings with controlled lighting and uncluttered backgrounds, conditions that rarely hold in a real home, and their false-alarm rates in naturalistic environments have been reported as uncomfortably high.

C. Deep-Learning Approaches for Activity Recognition

Convolutional networks and YOLO-family detectors have brought posture-level understanding within reach. Reported fall-recognition accuracy on standard benchmark datasets now regularly exceeds ninety per cent, and several authors have shown that providing a short temporal context—a window of consecutive frames rather than a single frame—further improves both precision and recall. The limitation that affects nearly all published work in this category is the deployment model: inference runs on a GPU server accessed over a network. Cloud-side processing adds transmission latency, requires a reliable broadband connection and sends raw video of a person's home to a third-party infrastructure, each of which is a barrier to adoption in practice.

D. Health-Parameter Monitoring

Heart rate and blood-oxygen saturation have been incorporated into smart-home platforms through ECG patches, PPG modules and consumer smartwatches. PPG sensors in particular have matured to where a coin-sized module costing a few dollars can match clinical-grade accuracy when correctly positioned. Despite this progress, most published systems treat posture monitoring and heart-rate monitoring as separate problems addressed by separate papers. A system that detects a fall but cannot simultaneously observe a coexisting tachycardia, or one that flags an abnormal heart rate but cannot check whether the resident has also collapsed, provides an incomplete picture of the resident's state. Fusing both modalities in a single decision layer is therefore an attractive design direction.

E. Edge Computing for Healthcare

Processing data near the point of capture rather than in a remote data center has attracted sustained interest in healthcare applications for three concrete reasons. Alert latency is reduced because there is no round-trip to an external server. The privacy exposure is smaller because sensitive data stays on the device. Network dependence is eliminated, which matters in the domestic environments where older adults live, where internet connectivity may be intermittent or absent. The Raspberry Pi 4, the NVIDIA Jetson Nano and the Coral Dev Board have all appeared as edge nodes in recent healthcare papers. We chose the Raspberry Pi because its software ecosystem is the most mature, replacement boards are available locally in Andhra Pradesh, and its cost makes large-scale deployment financially realistic.

F. Research Gap

From this review, five issues stand out as unresolved: wearable non-compliance, the semantic shallowness of threshold-based cameras, the cloud dependency of deep-learning systems, the absence of fused multi-modal decision logic, and the privacy implications of remote video transmission. Our system addresses all five directly by running YOLOv5 inference on a Raspberry Pi, combining posture analysis with on-board pulse sensing in a single rule layer, and transmitting nothing beyond compact e-mail notifications that contain no images.

III. PROPOSED METHOD

We organized the system around three functional layers. The input layer collects raw data: a USB camera captures video and a PPG sensor generates an analog signal proportional to blood-volume pulse. The processing layer is where the Raspberry Pi does its core work: running the YOLOv5 detector, processing ADC samples and evaluating alert conditions. The output layer acts on the processing layer's decisions by driving a buzzer and sending e-mail. Fig. 1 illustrates how these blocks connect. Power is supplied through a regulated 12 V adapter for the buzzer and sensor breakouts; the Raspberry Pi draws power through its USB Type-C connector on a separate supply rail so that inrush current from the buzzer cannot cause a voltage drop on the processor supply.

A. Raspberry Pi Processing Unit

The Raspberry Pi 4 Model B is the computational core of the platform. Its quad-core ARM Cortex-A72 processor clocked at 1.5 GHz is able to run the YOLOv5 nano model at three to five frames per second after the weights have been exported to ONNX format and loaded through OpenCV's DNN module. This frame rate comfortably exceeds what is needed to track posture changes that occur on a scale of seconds. The 40-pin GPIO header provides direct access to the SPI bus needed by the MCP3008 ADC, a digital output for the buzzer and a digital input for an optional infrared presence sensor near the doorway. The operating system and application run from a 16 GB microSD card.

B. USB Camera and Visual Capture

We used a standard USB webcam configured to deliver 640 × 480 pixel frames at fifteen frames per second. OpenCV's VideoCapture reads frames and passes them to the detection thread, where each frame is resized to 640 × 640 pixels and its pixel values normalized before being forwarded to YOLOv5. Choosing 640 × 480 as the capture resolution gave us enough spatial detail to distinguish standing from lying postures across a living-room-sized space while keeping preprocessing time short.

C. Heartbeat Sensor and MCP3008 Converter

Because the Raspberry Pi has no on-board ADC, we used an MCP3008 ten-bit successive-approximation converter that communicates over SPI and supports eight single-ended channels. Channel zero was connected to the output of a PPG finger-clip module; the remaining seven channels are available for future sensors. We polled the MCP3008 every ten milliseconds, producing a 100 Hz sample stream that is well above the Nyquist rate for the cardiac frequencies of interest. The connection assignments are summarized in Table I.

TABLE I. MCP3008 ADC to raspberry pi pin assignment.

MCP3008 Pin	Pin Name	Raspberry Pi Pin	GPIO Number
Pin 1	CH0 (Analog In)	Heartbeat Sensor Output	-
Pin 9	DGND	Pin 6 (GND)	GND
Pin 10	CS/SHDN	Pin 24	GPIO 8 (CE0)
Pin 11	DIN (MOSI)	Pin 19	GPIO 10 (MOSI)
Pin 12	DOUT (MISO)	Pin 21	GPIO 9 (MISO)
Pin 13	CLK	Pin 23	GPIO 11 (SCLK)
Pin 14	AGND	Pin 6 (GND)	GND
Pin 16	VDD	Pin 1 (3.3 V)	3.3 V Supply

D. Buzzer, IR Sensor and Power Path

A 5 V active buzzer connects through a 100 Ω current-limiting resistor to GPIO pin 17. Setting this pin high causes the buzzer to sound immediately. An optional passive-infrared sensor above the doorway can wake the system from a reduced-power mode in which YOLOv5 inference is suspended to conserve energy; in our tests we bypassed this sensor and ran inference continuously. Total power consumption under typical load was measured at under ten watts.

IV. PROPOSED ALGORITHM

A. Visual Pipeline

For each frame delivered by VideoCapture the pipeline resizes, normalizes and passes the image through YOLOv5, which returns a list of detections each carrying a class label, a confidence score and a pixel-coordinate bounding box. Detections with confidence below 0.45 are discarded; this threshold was settled on after indoor testing and represents a practical trade-off between noise suppression and reliable person detection under moderate lighting. For each surviving person detection we compute aspect ratio as bounding-box height divided by width. Values above 1.6 correspond to standing or walking, values between 0.9 and 1.6 to sitting, and values below 0.7 to a lying posture.

B. Posture-Based Decision Logic

A single frame with a low aspect ratio does not by itself trigger a fall alert. We require the ratio to remain below 0.7 for more than two consecutive seconds, which at our processing rate corresponds to at least six or seven successive frames. A second condition must also hold: the lower edge of the bounding box must cross a horizontal calibration line we drew on a reference image at installation time to mark floor level. This floor check prevents alerts when a person reclines on a sofa, whose bounding box may have a low aspect ratio but whose feet remain well above the floor.

Self-harm gestures—characterized by repeated movement of the hands toward the head—are detected by tracking the centroid of any small bounding box and counting how often it comes within a pixel-distance threshold of the top region of the person box within a five-second window. Continuous agitated movement is flagged through the temporal variance of the person centroid: when that variance exceeds an empirically set bound over several consecutive seconds the system raises an abnormal-behavior alert.

C. Physiological Pipeline

A second thread runs independently of the visual pipeline. It samples the MCP3008 every ten milliseconds, appending each reading to a ring buffer covering the most recent five seconds. At the start of each new second a single-pole high-pass filter removes slow DC drift from the buffer, and a peak-detection routine counts inter-peak intervals to produce a beats-per-minute estimate, which is placed in a thread-safe queue for the alert layer. BPM values below 60 or above 100, and readings whose inter-beat standard deviation exceeds a configurable bound, are flagged as physiologically abnormal.

D. Alert Layer

The alert layer reads both the visual result and the physiological queue at each processing cycle and passes them through a small state machine. When both channels simultaneously report an abnormality the event is treated as critical; when only one channel reports a problem it is treated as standard. Critical events activate the buzzer at full volume for ten seconds and trigger an immediate e-mail. Standard events trigger the e-mail only. A sixty-second cooldown between successive alerts of the same class prevents notification flooding during a sustained event. The e-mail is assembled using Python's smtplib over a STARTTLS-secured connection and contains the event class, local timestamp, most recent BPM and a short human-readable description.

TABLE II. Event types, trigger conditions, and responses.

Event Type	Visual Trigger	Physiological Trigger	Action
Fall	Vertical to horizontal posture > 2 s	Optional drop in BPM	Buzzer + email
Self-Harm	Hands-to-head gesture pattern	Possible BPM > 100	Buzzer + email
Abnormal Behavior	Continuous shaking > 2 s	Irregular BPM	Buzzer + email
Health-Only	Posture normal	BPM < 60 or > 100	Email alert

V. RESULTS AND DISCUSSION

A. Experimental Setup

We assembled the prototype on a base plate in our ECE laboratory at Aditya College of Engineering and ran thirty-seven scripted scenarios over two testing days. Normal-activity scenarios included standing, sitting, slow walking, simulated cooking and watching a screen. Emergency scenarios included controlled falls onto a padded mat, deliberate hand-to-head gestures, rhythmic shaking and short bouts of light exercise to elevate heart rate. Each scenario was run at least three times under daytime fluorescent lighting, single-table-lamp evening lighting and near-darkness with IR assistance. A second observer with a stopwatch recorded ground-truth event start and end times.

B. Fall Detection

Fig. 3 shows a representative fall trial screenshot. The left pane shows the YOLOv5 console log alongside the live BPM readout; the right pane shows the corresponding video frame at the instant the posture transition was classified. Across twenty controlled falls the model correctly identified eighteen, missed one where a chair partially occluded the subject, and generated one false positive when the subject performed an unusually slow lie-down that satisfied the aspect-ratio rule momentarily without triggering the floor-check condition. The two-second persistence window was the single most important parameter for suppressing false alarms caused by sitting cross-legged on a floor rug.

C. Self-Harm and Abnormal-Behavior Detection

Fig. 4 shows the detection output during a self-harm gesture trial. The hand-box centroid moved between the upper and middle thirds of the frame and crossed the head-proximity threshold twice within five seconds, satisfying the gesture rule. Fig. 5 shows an abnormal-behavior event triggered by a sudden forward lunge; the centroid variance over the preceding three seconds exceeded the threshold and the alert fired as expected. In both cases the visual log and simultaneous BPM reading were preserved in the console for auditing.

D. Health Monitoring and Alerts

Fig. 6 shows the e-mail received during a deliberate tachycardia trial. The subject line reads HEART RATE ALERT and the body reports 126 BPM along with a timestamp and a brief description. During resting conditions across all trials BPM estimates remained within the 60–100 BPM window. Brief artefacts caused by finger movement on the sensor were filtered reliably by the sliding-window peak detector, and the MCP3008 produced clean samples throughout the two testing days with no SPI errors.

E. Latency and False-Alarm Behavior

End-to-end latency from the first inciting frame to buzzer activation stayed below one second in every fall trial. The dual-channel design reduced false alarms noticeably: several brief posture ambiguities that would have triggered alerts in a vision-only system were suppressed because the heart-rate stream simultaneously showed nothing unusual. Table III summarizes the detection performance across all event categories.

TABLE III. Indicative event-level performance of the prototype.

Event Type	Trials	Correct	Missed	False Pos.
Fall	20	18	1	1
Self-Harm Gesture	12	11	1	0
Abnormal Behavior	10	9	1	0
Heart-Rate Alert	15	14	1	0

VI. CONCLUSION AND FUTURE SCOPE

This paper has presented and bench-tested a Raspberry Pi–based monitoring platform that performs YOLOv5 posture detection and PPG heart-rate estimation locally, without any cloud component. Experimental results across more than thirty scripted scenarios showed that bounding-box aspect-ratio analysis combined with temporal persistence checking and dual-channel decision logic reliably identifies falls, self-harm gestures and abnormal movement with low false-alarm rates and sub-second response times. Privacy is protected by construction because raw video never leaves the device.

Future work should address four areas. First, expanding the training data to cover more body types, clothing styles and room layouts would reduce the misses that occurred under partial occlusion and low light. Second, adding SpO_2 and non-contact temperature sensing would enrich the physiological channel beyond pulse rate alone. Third, MQTT-based integration with hospital information systems would allow genuine emergencies to be escalated automatically to medical staff. Fourth, and most importantly, evaluation in naturalistic conditions with real elderly volunteers is needed before any strong claims about clinical utility can be made. Taken together these steps would move the platform meaningfully closer to a deployable assistive technology for independent living.

ACKNOWLEDGEMENT

The authors acknowledge the Department of Electronics and Communication Engineering, Aditya College of Engineering, Madanapalle, for access to the laboratory facilities used in this work, and thank the Project Coordinator and the reviewers whose comments improved the quality of the paper.

REFERENCES

[1] World Health Organization, World Report on Ageing and Health, Geneva, Switzerland, 2015.
[2] G. Jocher et al., YOLOv5: A State-of-the-Art Real-Time Object Detector, Ultralytics GitHub Repository, 2020.
[3] J. Redmon and A. Farhadi, YOLOv3: An Incremental Improvement, arXiv:1804.02767, 2018.
[4] X. Yu, Approaches and Principles of Fall Detection for Elderly and Patient, Proc. IEEE HealthCom, pp. 42–47, 2008.
[5] M. Mubashir, L. Shao and L. Seed, A Survey on Fall Detection: Principles and Approaches, Neurocomputing, vol. 100, pp. 144–152, 2013.
[6] N. Noury, A. Fleury, P. Rumeau et al., Fall Detection – Principles and Methods, Proc. IEEE EMBC, pp. 1663–1666, 2007.
[7] J. Allen, Photoplethysmography and Its Application in Clinical Physiological Measurement, Physiological Measurement, vol. 28, no. 3, pp. R1–R39, 2007.
[8] Microchip Technology Inc., MCP3004/3008 2.7 V 4-Channel/8-Channel 10-Bit A/D Converters with SPI Serial Interface, Datasheet, 2008.
[9] Raspberry Pi Foundation, Raspberry Pi 4 Model B Product Brief, 2019.
[10] S. Ren, K. He, R. Girshick and J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, 2017.
[11] A. Bochkovskiy, C.-Y. Wang and H.-Y. M. Liao, YOLOv4: Optimal Speed and Accuracy of Object Detection, arXiv:2004.10934, 2020.
[12] Y. Liu, B. Wang and W. Wei, Edge Computing for Internet of Things: A Survey, IEEE Access, vol. 8, pp. 116903–116917, 2020.
[13] M. Chen, Y. Ma, Y. Li, D. Wu, Y. Zhang and C.-H. Youn, Wearable 2.0: Enabling Human-Cloud Integration in Next Generation Healthcare Systems, IEEE Commun. Mag., vol. 55, no. 1, pp. 54–61, 2017.
[14] K. Chen, R. Zhang, J. Zhao and X. Zhang, A Real-Time Human Activity Recognition Approach Using Smartphones, Sensors, vol. 18, no. 9, p. 3036, 2018.
[15] S. R. Steinhubl, E. D. Muse and E. J. Topol, The Emerging Field of Mobile Health, Science Translational Medicine, vol. 7, no. 283, p. 283rv3, 2015.
[16] P. Viola and M. Jones, Robust Real-Time Face Detection, Int. J. Comput. Vis., vol. 57, no. 2, pp. 137–154, 2004.
[17] A. Krizhevsky, I. Sutskever and G. E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, Commun. ACM, vol. 60, no. 6, pp. 84–90, 2017.
[18] K. He, X. Zhang, S. Ren and J. Sun, Deep Residual Learning for Image Recognition, Proc. IEEE CVPR, pp. 770–778, 2016.