Sitting Posture Corrector App on Raspberry Pi Using Edge AI

A realtime posture correction system

Pranav, Shreyas, Ashwin, Ronalda, Veningston

ACM India Winter School on Edge AI (Hackathon) @ IISc, Bangalore

January 4, 2026

Problem

- Prolonged poor sitting posture (neck-down, slouch, leaning) increases discomfort and lead to severe health hazard.
- Goal: Detect sitting posture in real time and provide **actionable feedback** using a **0–100 score** and **warning**.

Proposed Solution

- Works **on-device**
- Single camera in a **side view**.
- **MoveNet (TFLite)** estimates 17 body keypoints in real time.
- A calibrated scoring module maps posture deviation to **Score** $\in [0, 100]$.
- Raises warning to indicate **Good / Bad** posture.
- Neck-down / forward-head posture receives higher weight than mild scouch and leaning.

System Overview

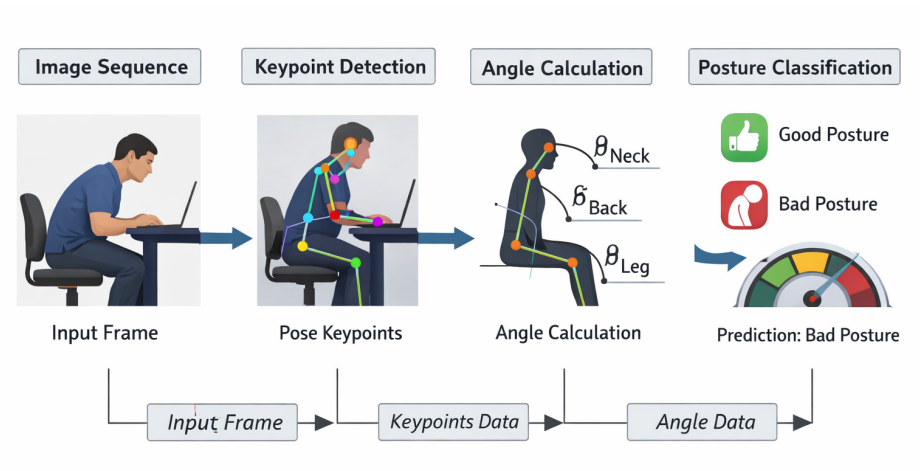


Figure 1: End-to-end pipeline for posture monitoring in side view: input video frames → MoveNet keypoint extraction → angle computation → posture prediction and score visualization.

Hardware & Deployment

- **Edge device:** Raspberry Pi (camera)
- **Compute:** On-device inference with `tflite_runtime`
- **Privacy:** Video stays local; only score/warning output leaves the pipeline

Pose Estimation with MoveNet

MoveNet outputs keypoints

$$\text{KP} = \{(x_i, y_i, c_i\}_{i=1}^{17}$$

where, x_i, y_i, c_i are normalized coordinates and confidence. Interest points from head, shoulder, hip.

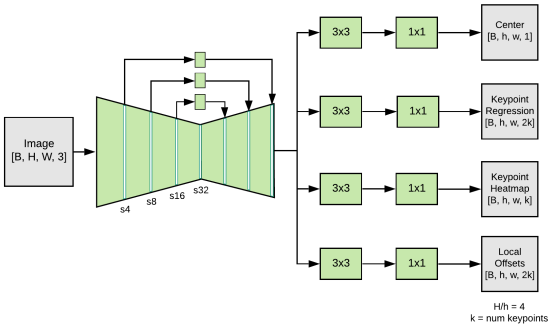


Figure 2: MoveNet Architecture

Duration Gating & Warning

Short movements should not trigger warnings.

Duration gating (per feature)

- Mild deviation: count if sustained $\geq$ 8–10s
- Moderate deviation: sustained $\geq$ 3–5s
- Severe deviation: sustained $\geq$ 1–2s

ALERT after hold time

- Score ≥ 80: Green (Good)
- $60 \leq$ Score < 80: Yellow (Warning)
- Score < 60: Red (Bad posture)

Algorithm Outline

1. Capture frame, preprocess to MoveNet input size.
2. Run MoveNet inference $\rightarrow$ keypoints $\{(x_i, y_i, c_i)\}$.
3. Apply confidence gating; compute head/shoulder/hip midpoints.
4. Extract angle features: neck flexion, trunk flexion, forward-head proxy, lean/asymmetry.
5. Compute penalties p_f and Score $\in [0, 100]$.
6. Drive ALERT based on Score and hold time.

Fail-safe: if keypoint confidence drops, freeze score and do not penalize.

Experimental Evaluation

- **Posture detection quality**
  - Frame-level classification: Good vs Bad (and optional multi-class).
  - Confusion matrix, Accuracy, Precision, Recall, F1 (Bad = positive).
- **Score validity**
  - Collect scores for Neutral, Mild slouch, Severe slouch, Neck down, Lean.
  - Report mean $\pm$ std; boxplot by posture type.
- **Edge performance**
  - FPS and latency breakdown (preprocess + inference + scoring + warning).
  - Peak RAM and CPU utilization over 5 minutes.
- **Failure analysis**
  - False alarms per minute (or per hour) and typical failure cases.

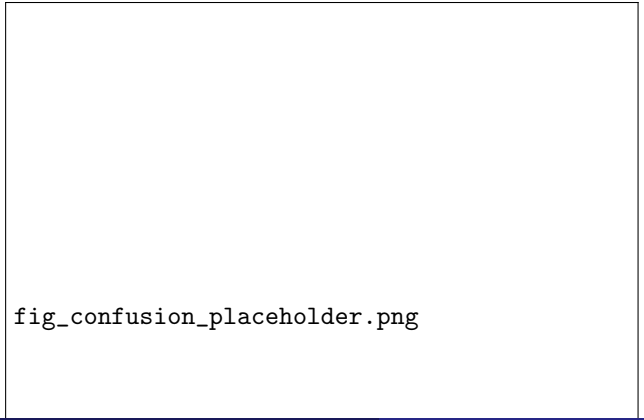
Posture Detection Quality (Dataset & Labeling)

- Labeled set: **15–30 minutes** across **5 participants**.
- Two human annotators provide:
  - Binary labels: Good / Bad
  - Optional multi-class: Good / Slouch / Neck-down / Lean
- Evaluation uses held-out segments (or cross-participant split).

Detection Metrics & Confusion Matrix

Binary metrics (Bad = positive):

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}, \quad F1 = \frac{2PR}{P + R}$$



- Report Accuracy, Precision, Recall, F1.
- Also report false alarms/min as a user-centric metric.

Score Validity (Controlled Postures)

- Collect short trials for each condition:
  - Neutral good posture
  - Mild slouch
  - Severe slouch
  - Neck down (high priority)
  - Lean (left/right)
- Report mean $\pm$ std score per posture type.
- Visualize using boxplots to show separation and variance.

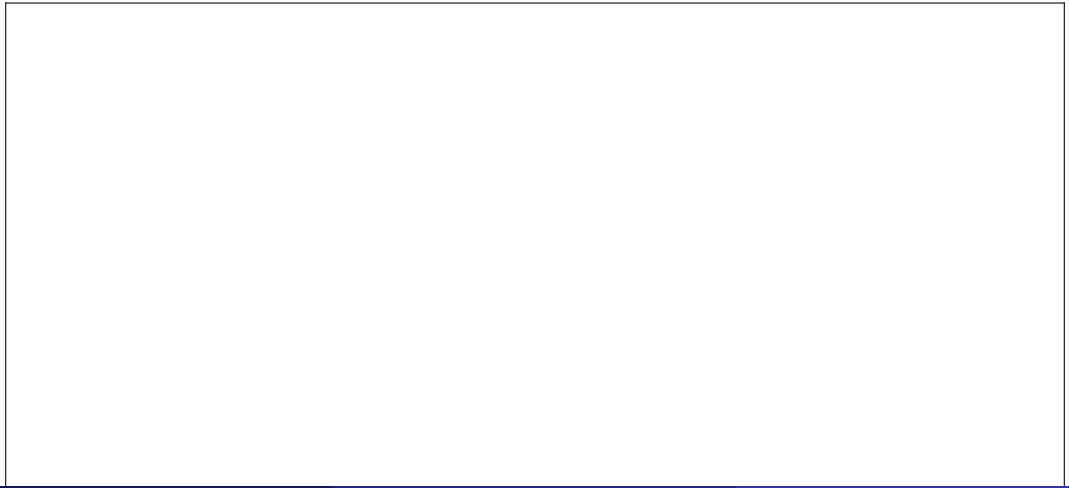
Edge Performance (FPS & Latency)

- End-to-end FPS: camera $\rightarrow$ inference $\rightarrow$ scoring $\rightarrow$ Alert
- Latency breakdown (ms):
  - preprocessing
  - inference
  - feature + scoring
  - alert update
- Test at multiple input resolutions (if feasible).

Input Size	FPS	Inference (ms)	Total (ms)
192×192	–	–	–
256×256	–	–	–
320×320	–	–	–

Resource Footprint (CPU & RAM)

- Measure CPU utilization and RAM usage over 5 minutes.
- Report peak RAM and average CPU load.



Failure Analysis

- Report false alarms per minute (or per hour).
- Typical failure modes:
  - low light / motion blur
  - occlusion (face/shoulders not visible)
  - unusual seating position or camera moved after calibration
- Mitigation:
  - confidence gating + freeze score
  - re-calibration prompt when baseline shifts
  - duration gating to suppress brief movements

Screenshots

- Camera feed with pose overlay (keypoints + skeleton).
- Real-time Score (0–100).
- ALERT response: Green / Yellow / Red with hold time.
- Example scenarios: neutral vs neck-down vs slouch vs lean.

Conclusion

- EdgeAI posture corrector runs locally on Raspberry Pi with real-time feedback.
- Calibration enables robust scoring under a single camera view.
- Neck-down posture is prioritized using weighted penalties and duration gating.

Future Work

- Multi-view posture estimation (for true 3D triangulation).
- Upgrade from single-person MoveNet to a multi-person pose pipeline (detect persons $\rightarrow$ run pose per ROI)
- Expand classes (e.g., asymmetric sitting, phone neck).

Thank you!

Questions?