

Sitting Posture Corrector App on Raspberry Pi Using Edge AI

A realtime posture correction system

Pranav, Shreyas, Ashwin, Ronald, 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 **isometric/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** indicate **Good / Warning / Bad** posture.
- Neck-down / forward-head posture receives higher weight than mild scouch and leaning.

System Overview

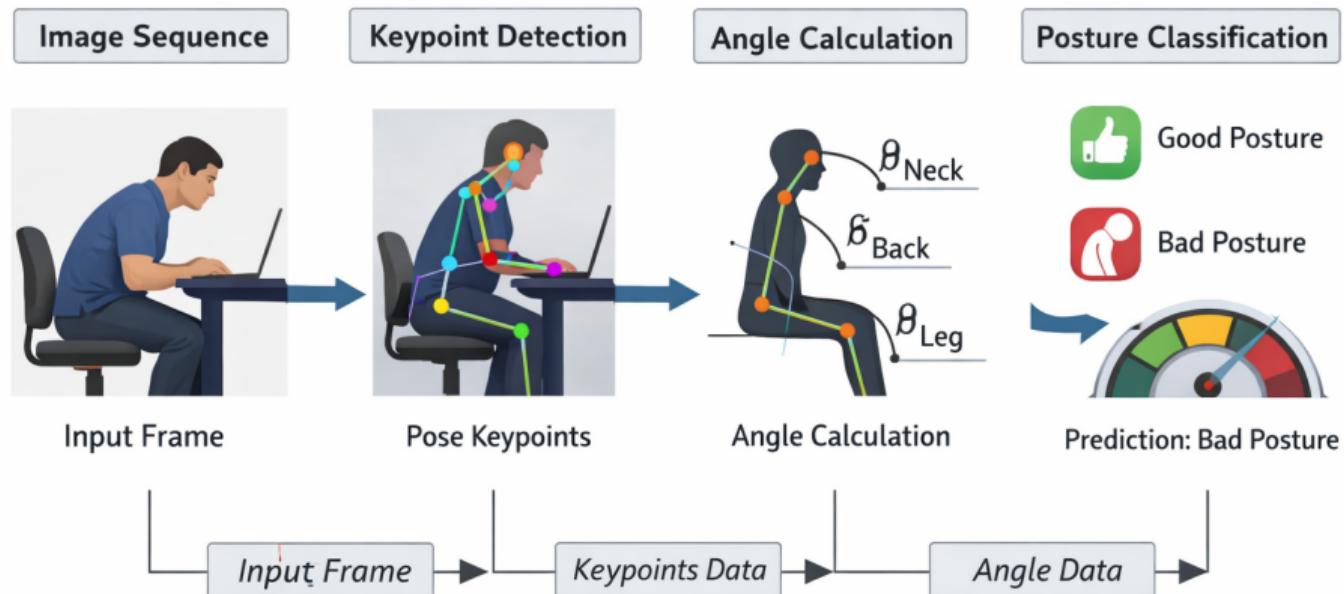


Figure 1: End-to-end pipeline for posture monitoring in side view: input video frames → MoveNet keypoint extraction → angle computation → posture prediction and score visualization.

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

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

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

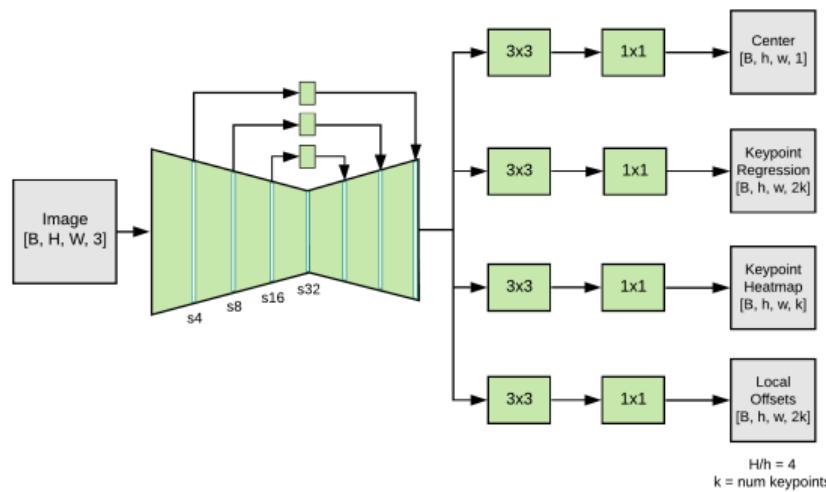


Figure 2: MoveNet Architecture

Calibration

- Collect 5–10 seconds of **neutral good posture**.
- Compute baseline statistics for each feature:

$$\mu_f = \mathbb{E}[f], \quad \sigma_f = \sqrt{\mathbb{E}[(f - \mu_f)^2]}$$

- Runtime scoring uses **deviation from baseline**:

$$\Delta f = f_t - \mu_f \quad (\text{or } |\Delta f| \text{ depending on feature})$$

- This reduces sensitivity to camera angle and user-specific anatomy.

Posture Features from Keypoints

Define midpoints:

$$\mathbf{s} = \frac{\mathbf{s}_L + \mathbf{s}_R}{2}, \quad \mathbf{h} = \frac{\mathbf{h}_L + \mathbf{h}_R}{2}$$

and torso length:

$$L = \|\mathbf{s} - \mathbf{h}\|$$

Key features:

- **Neck-down (highest weight):** *neck flexion proxy*

$$\theta_{\text{neck}} = 180^\circ - \angle((\mathbf{p} - \mathbf{s}), (\mathbf{h} - \mathbf{s}))$$

where \mathbf{p} is head point (nose → eyes → ears fallback).

- **Trunk slouch:** angle of $(\mathbf{s} - \mathbf{h})$ vs. vertical.
- **Forward-head proxy:** normalized horizontal offset

$$f_{\text{fwd}} = \frac{p_x - s_x}{L}$$

- **Lean & asymmetry:** head roll and shoulder height difference.

Scoring (0–100) with Neck Priority

Each feature produces a penalty $p_f \in [0, 1]$ using thresholds and duration gating.

Weighted badness:

$$B = \sum_f w_f p_f \quad \text{with} \quad \sum_f w_f = 1$$

Score:

$$\text{Score} = 100 \cdot (1 - B), \quad \text{clamp to } [0, 100]$$

Neck prioritized weights:

Feature	Symbol	Weight
Neck-down / neck flexion	θ_{neck}	0.40
Forward head proxy	f_{fwd}	0.25
Trunk slouch	θ_{trunk}	0.20
Lean + shoulder asymmetry	(combined)	0.15

Duration Gating & Warning

Short movements should not trigger **warnings**.

Duration gating (per feature)

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

ALERT after hold time

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

Algorithm Outline

- ① Capture frame, preprocess to MoveNet input size.
- ② Run MoveNet inference → 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.

fig_confusion_placeholder.png

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 → inference → scoring → 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 → run pose per ROI)
- Expand classes (e.g., asymmetric sitting, phone neck).

Thank you!
Questions?