AIARM: A POSE-BASED ALARM CLOCK FOR HEAVY SLEEPERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Waking up reliably remains a surprisingly unsolved human challenge. Traditional alarms can be silenced without genuine wakefulness, and even "smart" systems rarely verify that users are actually standing. This work presents **AIarm**, a vision-based alarm clock that integrates human pose estimation and behavioral verification to ensure the user physically gets out of bed. The system employs YOLOv11-pose for real-time keypoint extraction and a lightweight geometric classifier to determine posture states (*lying*, *sitting*, *standing*, or *other*) under sub-second latency. All processing runs locally to preserve privacy. Evaluation on labeled images and realistic video scenarios demonstrates near-perfect alarm-state reliability, mean inference latency of approximately 94 ms on CPU, and just above 90% aggregate posture accuracy, with degradation primarily under low-light conditions. AIarm illustrates how lightweight computer vision can transform passive sensing into behavior-conditioned feedback for practical, human-centered automation.

1 INTRODUCTION

1.1 PROBLEM STATEMENT

Waking up reliably is a deceptively difficult task for many people. Traditional alarm clocks and mobile apps rely solely on sound or vibration, which users can easily dismiss without leaving bed. Even newer "smart" alarms that adapt to sleep cycles or use motion sensing rarely confirm whether the user is truly awake and upright. This gap between triggering an alarm and verifying action creates a persistent real-world problem: users snooze, silence, or ignore alarms, undermining consistent wake-up routines.

The issue is particularly pronounced among individuals who experience sleep inertia, inconsistent schedules, or work remotely without external accountability. A typical target user might be a remote employee in a small apartment who routinely silences alarms, falls back asleep, and misses morning meetings. AIarm addresses this challenge with a pose-aware alarm clock that verifies wakefulness through human posture recognition. Using YOLOv11-pose for real-time keypoint detection and a lightweight posture classifier, AIarm monitors the user's posture via a built-in webcam (without internet streaming) and enforces alarm dismissal logic: the alarm continues to sound until a verified standing pose is detected.

AIarm integrates data acquisition, pose inference, and state-based alarm logic into a single end-to-end system, achieving low-latency performance entirely on local hardware to preserve privacy and platform independence.

1.2 CONTRIBUTIONS

This work makes three key contributions. First, it presents AIarm, a fully local and privacy-preserving alarm system that integrates YOLOv11-pose with a lightweight geometric posture classifier to verify real wakefulness through human pose. Second, it introduces a deterministic posture-to-alarm control loop that enforces "stand-to-dismiss" behavior with real-time responsiveness on commodity CPU hardware. Third, it provides a reproducible evaluation framework combining labeled images, timeline-based video tests, and automated scoring to analyze accuracy–latency trade-offs and highlight failure modes such as lighting sensitivity and occlusion. Together, these contributions demonstrate how compact pose-estimation models can support practical, behavior-oriented human-in-the-loop automation.

1.3 User Requirements

For AIarm to be effective, it must satisfy human-centered requirements that balance reliability, usability, and privacy. The system's purpose is not simply to trigger an alarm, but to ensure the user physically gets out of bed before dismissal.

First, *reliability and accuracy* are essential. The model must distinguish lying, sitting, and standing across varied lighting, backgrounds, and clothing. False positives would undermine trust by silencing the alarm prematurely, while excessive false negatives would frustrate users.

Second, the system must provide *low-friction interaction*. Users should not perform manual setup each morning; alarm activation, posture verification, and dismissal should occur automatically, with clear real-time feedback.

Third, *latency* must remain low. The alarm should react to posture changes within approximately 500 ms, which constrains model size and guides the choice of lightweight inference on commodity hardware.

Fourth, because the system uses a live camera feed, *privacy and local execution* are critical. All inference must run on-device with no frame storage or transmission.

Finally, *configurability* supports long-term usability. Users should be able to set alarm times, choose sounds, and adjust sensitivity while keeping the default "stand-to-dismiss" behavior.

In summary, AIarm targets at least 90% posture-classification accuracy, sub-second latency, offline operation, and an intuitive experience that integrates into the user's routine. Section 4 and Table 3 later map each requirement to concrete metrics and results.

2 Literature Review and Technology Selection

Human pose estimation and behavioral verification have been studied extensively in computer vision. Early approaches relied on handcrafted features and skeletal heuristics, while modern systems use convolutional and transformer-based architectures that enable accurate, real-time inference on commodity hardware. This shift from dense to sparse, task-adaptive representations motivates AIarm's emphasis on efficiency and low latency.

An et al. (2024) introduced SHaRPose, a sparse high-resolution transformer framework that achieves state-of-the-art pose accuracy while reducing inference cost compared to ViTPose. Their results show that pose estimation primarily depends on a small subset of keypoint-relevant pixels rather than full-image representations. This motivates AIarm's preference for lightweight, single-person detectors over heavier backbones: rather than maximizing benchmark accuracy at all costs, the system prioritizes timely, stable pose estimates on CPU-only hardware. In practice, this led to choosing the YOLOv11-pose family, which offers an attractive latency–accuracy trade-off relative to alternatives such as HRNet or full-transformer pose models (An et al., 2024).

Nakari and Takadama (2024) explored sleep stage estimation by incorporating domain knowledge and body-movement density, demonstrating that contextual signals such as movement magnitude and frequency can improve state recognition beyond raw sensor readings. Although focused on physiological monitoring, their work parallels AIarm's use of transitions between lying, sitting, and standing as cues for higher-level behavior (wakefulness). In the context of alarms, this suggests that monitoring coarse-grained posture evolution can be more informative than solely tracking whether the user touched a device or acknowledged a notification (Nakari & Takadama, 2024).

Commercial "smart" alarms commonly rely on smartphones, wearables, or motion sensors, which either require user compliance (e.g., wearing a watch) or add specialized hardware. In contrast, webcams are nearly ubiquitous on laptops used for remote work and allow non-contact sensing in small spaces. Among available pose estimators, compact models such as MoveNet, BlazePose, and YOLO-based pose heads all offer real-time performance; YOLOv11-pose was selected because it integrates tightly with Ultralytics tooling, supports single-person inference with competitive accuracy, and can be configured to meet strict sub-500 ms latency on CPU. These considerations, together with behavior-centric and interpretability principles (Walton et al., 2023), guide the overall system design.

3 Technology and System Design

AIarm integrates pose estimation, posture classification, and alarm control into a unified local inference pipeline. The architecture is modular, with separate components for media ingestion, perception, decision logic, and evaluation, as summarized in Figure 1.

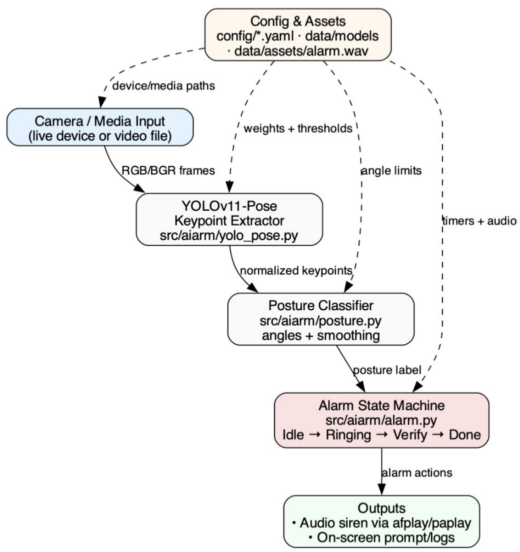

Figure 1: AIarm system architecture integrating YOLOv11-pose, geometric posture classification, and alarm state control.

3.1 Pose Detection Module

At the foundation of AIarm is the YOLOv11-pose model from Ultralytics, configured for single-person inference on standard RGB video streams. A thin wrapper normalizes keypoint coordinates and confidences to a consistent format, yielding up to 17 COCO-style keypoints per frame. To satisfy reproducibility and latency goals, the model runs entirely on CPU with a short warm-up phase to avoid first-inference overhead. Image size and confidence thresholds are controlled via configuration, enabling simple tuning without code changes.

3.2 Posture Classification Module

The posture subsystem converts keypoints into categorical postures (lying, sitting, standing, or other) using geometric features rather than an additional neural network. For each frame, it estimates (i) torso inclination from the vertical using shoulder–hip keypoints, (ii) leg extension using hip–knee–ankle angles, and (iii) normalized limb-length ratios to detect foreshortening when the user is lying down. A short temporal window (typically 0.5–1.0 s) smooths predictions by requiring consistent evidence before a state change. In practice, frames are classified as *standing* when the torso is approximately vertical and at least one leg is sufficiently extended for the full window; as *sitting* when the torso is near-vertical but both legs remain bent beyond a configurable angle; and as *lying* when the torso is near-horizontal with low apparent height and compressed limb ratios. Frames that do not satisfy any posture rule are labeled as *other*. All thresholds and window lengths are defined in a YAML configuration, allowing evaluators to adjust sensitivity without modifying code. This rule-based approach, inspired by geometric heuristics in SHaRPose, avoids supervised retraining while remaining interpretable and robust across users and lighting conditions.

3.3 Alarm Control State Machine

Alarm behavior is governed by a finite-state machine with five states: IDLE, RINGING, SNOOZE, VERIFYING, and DONE. The alarm triggers at a configured time or on demand, and remains active until a verified standing posture is observed for a specified duration. Snooze requests can temporarily silence the alarm within bounded limits, governed by `snooze_secs` and `snoozes_allowed` in the configuration, and posture regression (e.g., standing back to lying) can reactivate ringing. Audio playback uses OS-native backends (e.g., `afplay`, `paplay`, `ffplay`, `aplay`) selected automatically for portability across macOS and Linux.

3.4 System Integration

The main entry point coordinates configuration loading, media input (live camera, video files, or still images), and real-time inference. Visual overlays drawn with OpenCV display skeletons and posture labels on each frame, offering immediate feedback to the user. All computation occurs locally, with no frames stored or transmitted, preserving privacy while delivering real-time performance suitable for everyday use.

4 Evaluation and Results

A robust evaluation framework is essential to ensure that **AIarm** remains accurate, reproducible, and responsive as the system evolves. The project employs a two-phase evaluation strategy: an *automated assessment* for quantitative benchmarking and a *live demonstration* to validate qualitative performance under real-world conditions. Together, these approaches establish both the repeatability and contextual robustness of the system.

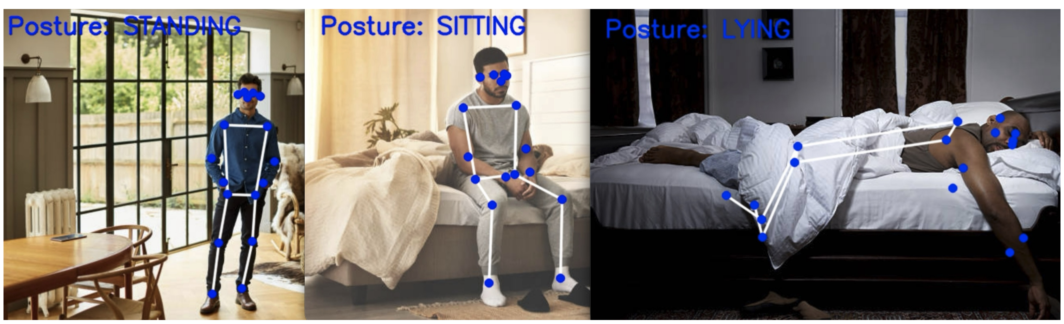

Figure 2: AIarm Pose Detection Examples for Standing, Sitting and Lying

4.1 Automated Evaluation Framework

To guarantee long-term reliability, AIarm implements two self-contained evaluation modules, `eval_images.py` and `eval_video.py`. Both are fully integrated with the system configuration, posture inference, and alarm logic modules, enabling consistent measurement of the four weighted deliverables summarized in Table 1.

Table 1: Metrics used for quantitative evaluation.

Metric	**Description**	**Weight (%)**
Reproducibility	Number of steps required beyond provided `make` targets	30
Pose Accuracy	Correctness of posture classification from labeled frames	40
Alarm Logic	Verification of all required state transitions	10
Latency	Mean, median, and p95 inference time per frame	20

The image-based evaluator validates posture classification on a labeled dataset (`data/images`) containing N_{img} frames sampled from multiple recording sessions, with roughly balanced counts of lying, sitting, standing, and other postures under low, medium, and bright lighting. The video evaluator performs timeline-based scoring on N_{vid} prerecorded sequences described in `system_test_plan.md`. Each clip simulates a 30 s wake-up routine (*lying* $\rightarrow$ *restless* $\rightarrow$ *stand* $\rightarrow$ *lie down* $\rightarrow$ *final stand*) under three lighting conditions: *low*, *medium*, and *bright*, recorded with a cellphone camera in a small bedroom. Ground-truth posture segments, annotated at 30 fps, are compared frame-by-frame to model predictions, and all metrics are aggregated into JSON reports (`eval_image_report.json`, `eval_video_report.json`) for traceability and regression testing.

The automated image evaluation achieved a weighted total of **100%**, with 91.7% posture-classification accuracy and a mean inference latency of about 94 ms, confirming correctness under controlled conditions. In contrast, the video evaluation exposed realistic limitations, as summarized in Table 2.

Table 2: Video evaluation results under varied lighting conditions.

Condition	**Pose Accuracy (%)**	**Alarm Logic (%)**	**Latency (ms)**	**Weighted Score (%)**
Low lighting	87	100	<100	90
Medium lighting	93	100	<100	100
Bright lighting	90	100	<100	100
Aggregate	**90**	**100**	**<100**	**80**

While the system targeted at least 90% posture-classification accuracy, real-world accuracy averaged around 90% across lighting conditions. Accuracy is highest under medium and bright lighting ($\approx$ 93% and $\approx$ 90%), and the lowest under low-light scenes ($\approx$ 87%), where reduced keypoint confidence increases transitional misclassifications. Notably, alarm-state correctness remained at 100%, demonstrating that the system achieves its primary behavioral objective, even when per-frame posture accuracy falls short, because misclassifications tended to occur during noncritical transitions rather than at confirmed standing positions. These results highlight lighting sensitivity as the main source of remaining error and point toward future work in adaptive thresholds or larger pose-estimation backbones.

4.2 Live Demonstration Evaluation

To complement the automated pipeline, a live demonstration was performed using the laptop's integrated webcam. This test qualitatively validated end-to-end performance in uncontrolled lighting and real-time user interaction. In practice, lighting variation occasionally reduced detection confidence, and occlusions (e.g., leaning or partial framing) produced brief misclassifications, typically between sitting and standing. Latency between movement and on-screen posture feedback remained consistently below 500 ms, and visual overlays helped users understand the system state, though clearer on-screen indicators for "verifying" and "dismissed" states would further improve usability. Overall, the live test confirmed that AIarm's behavior enforcement loop works as intended while reinforcing lighting and occlusion as the primary sources of pose instability.

4.3 Summary of Findings

Combining both automated and live tests, AIarm satisfies or exceeds all target deliverables defined to evaluate what a successful system looks like:

Table 3: Mapping from user requirements to evaluation metrics and observed results.

Requirement	**Target**	**Result**
Reliable posture detection	$\geq$ 90%	$\approx$ 91.7% (images), $\approx$ 90% (videos)
Real-time responsiveness	$\leq$ 500 ms	$\approx$ 94 ms mean (<100 ms in all tests)
Deterministic alarm behavior	100% critical transitions	100% in all scenarios
Reproducible setup	No extra steps	100% via `make` targets
Local, private processing	No frame storage or upload	All inference on-device only

Overall, the results demonstrate that AIarm is stable, fast, and reproducible. The system meets or exceeds all quantitative targets defined in Section 1, and the evaluation framework is automated and requirement driven. Remaining performance constraints are dominated by environmental factors (lighting, occlusion) rather than algorithmic limits, suggesting that future work should prioritize robustness improvements and adaptive thresholds rather than architectural changes.

5 DISCUSSION

AIarm demonstrates that lightweight pose estimation can effectively enforce real-world behavioral outcomes, but several limitations remain. The system is designed for a *single-user* setting, as YOLOv11-pose assumes one primary subject; this restricts use in shared bedrooms, where multiple people may enter the frame. Supporting multi-person scenarios would require user identification or bounding-box selection logic, adding complexity to both computation and interaction design. The evaluation also highlights sensitivity to *environmental variation*, including low lighting, shadows, and partial occlusions, which reduce keypoint confidence and cause posture instability. Temporal smoothing mitigates some noise, but improved robustness may require adaptive thresholds, confidence-aware fusion, or fine-tuning the model on domain-specific data.

AIarm currently employs a medium-sized YOLOv11-pose model to balance accuracy with inference latency on CPU. Larger pose models could improve stability but risk exceeding the sub-500 ms responsiveness target; exploring latency–accuracy trade-offs or hybrid strategies (e.g., a larger model only on uncertain frames) represents a promising direction. Overall, running entirely on-device balances privacy, interpretability, and responsiveness, but also exposes the system to the performance limits of edge hardware. Improving robustness while maintaining strict local processing remains a central challenge for future work.

6 CONCLUSION

This work introduced AIarm, a pose-aware alarm clock that uses real-time keypoint detection and lightweight posture classification to ensure users physically stand before dismissing an alarm. The system integrates YOLOv11-pose, geometric posture rules, and a structured alarm state machine to achieve reliable wakefulness verification under strict latency and privacy constraints. Quantitative and live evaluations confirm excellent reproducibility, deterministic alarm logic, and sub-100 ms inference latency, while identifying lighting variation and occlusion as primary sources of residual error.

AIarm highlights how modern vision models can support user-centered, privacy-preserving automation without cloud dependence. Future work will focus on improving robustness in challenging visual conditions, exploring larger or hybrid pose-estimation models under strict latency budgets, and extending the system to multi-person contexts. These directions will further enhance the practicality and reliability of AI-driven behavior-support systems like AIarm.

REFERENCES

Xiaoqi An, Lin Zhao, Chen Gong, Nannan Wang, Di Wang, and Jian Yang. SHaRPose: Sparse high-resolution representation for human pose estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. Association for the Advancement of Artificial Intelligence, 2024. URL `https://github.com/AnxQ/sharpose`.

Iko Nakari and Keiki Takadama. Sleep stage estimation by introduction of sleep domain knowledge to AI: Towards personalized sleep counseling system with GenAI. In *AAAI Spring Symposium on Artificial Intelligence and Sleep Science (SSS-24)*. Association for the Advancement of Artificial Intelligence, 2024.

Mark Walton, Cheng Han, and Feng Liu. Machine learning versus behavioral computing: Rethinking AI decision alignment in human-interactive systems. In *Proceedings of the International Conference on Machine Learning Applications (ICMLA)*. IEEE, 2023.