Cerdika: Jurnal Ilmiah Indonesia, 6 (8), August 2026
p-ISSN: 2774-6291 e-ISSN: 2774-6534
Available online at http://cerdika.publikasiindonesia.id/index.php/cerdika/index

Implementation of a Quality Management System Framework for Optimizing Energy Efficiency of on-Device AI Inference on Android Devices

Billy Josef Waworuntu[1], Gerald Imanuel Palar[2], Marvein Christo Atu Sasikome[3], Yoan Virginia Kaesang[4], Cindra Caesaria Hamzah[5]
Politeknik Negeri Manado, Indonesia
Email: waworuntu.billy@gmail.com[1], palargerald07@gmail.com[2], marveinsasikome71@gmail.com[3], kaesangyoan@gmail.com[4], cindra.caesaria27@gmail.com[5]

Keywords:
Quality Management System;
PDCA;
on-device AI;
Android;
energy efficiency

Abstract

The rapid development of on-device artificial intelligence (AI) has increased the demand for efficient AI inference on mobile devices, particularly Android devices with limited computational resources, battery capacity, and thermal constraints. Although various optimization techniques, such as quantization, pruning, and adaptive inference, have been proposed, inconsistent evaluation procedures remain a significant challenge to achieving reproducible and comparable measurements of energy efficiency. This study aims to implement a Quality Management System (QMS) framework based on the Plan-Do-Check-Act (PDCA) cycle to systematically optimize and evaluate the energy efficiency of on-device AI inference while maintaining inference performance and model quality. The research employed a design study and experimental approach by deploying lightweight AI models for human presence detection and keyword spotting. The models were optimized using 8-bit post-training quantization, structured channel pruning, and early-exit mechanisms, followed by standardized measurements of energy consumption, inference latency, accuracy, model size, memory usage, and the energy-delay product (EDP) on Android devices. The results indicate that QMS-guided optimization provides a structured framework for improving energy efficiency by reducing computational workload and memory traffic while preserving inference reliability. In particular, post-training quantization combined with hardware acceleration reduced energy consumption, whereas adaptive inference mechanisms improved responsiveness under varying battery and thermal conditions. In conclusion, integrating QMS principles with on-device AI optimization establishes a reproducible, traceable, and continually improvable framework for developing energy-efficient AI applications on Android devices.

INTRODUCTION

On-device artificial intelligence (AI) offers enhanced privacy, low latency, and resilience to network disruptions. As mobile devices have become increasingly capable, interest has grown in deploying AI models directly on smartphones and tablets rather than relying on cloud-based inference. This shift has been driven by several considerations: privacy-sensitive applications benefit from keeping data on the device; latency-critical use cases require inference times that cloud round trips cannot consistently achieve; and connectivity-constrained environments require AI functionality that operates independently of network access. However,

on resource-constrained Android devices, energy consumption and performance stability remain significant challenges, particularly under high thermal loads and low-battery conditions.

Various techniques have been developed to address these challenges, including compute-efficient neural network architectures, such as MobileNet and EfficientNet (Mi, Feng, & Huang, 2022); quantization, which reduces the precision of weights and activations from 32-bit floating-point to 8-bit integer representations; pruning, which removes redundant network connections; knowledge distillation, in which smaller models are trained to replicate the behavior of larger models; and adaptive inference, which dynamically adjusts computational workloads according to input complexity or device state (Marinó, Petrini, Malchiodi, & Frasca, 2023). Although these techniques offer substantial potential for improving energy efficiency, their practical effectiveness depends heavily on the underlying hardware, runtime environment, and application context. Furthermore, energy-efficiency evaluation methodologies have not yet been standardized across the research community, making it difficult to compare results across devices and studies.

This inconsistency in evaluation methodology represents a significant gap in the on-device AI literature. Without standardized measurement protocols, reported improvements in energy efficiency may reflect differences in experimental design rather than genuine algorithmic advances. This study addresses this gap by implementing a framework based on a Quality Management System (QMS), ensuring that the design, measurement, evaluation, and continual improvement processes are conducted systematically and remain fully traceable. By applying the same rigor to AI system evaluation that ISO 9001 applies to quality management, this research establishes a foundation for reproducible, transparent, and comparable on-device AI research.

The integration of QMS principles with on-device AI development is particularly relevant in higher education laboratory environments, where reproducibility, documentation, and structured experimentation are essential. Although university laboratories often lack access to industrial-grade testing equipment and large-scale device fleets, they can still conduct rigorous research through carefully designed and consistently implemented experimental protocols. Accordingly, this study includes practical research artifacts, including baseline models, model compression workflows, test applications, and analysis scripts, enabling students and researchers to replicate experiments on their own Android devices and generate comparable quantitative results. This approach promotes experimental reproducibility, scientific rigor, and continual improvement in on-device AI research.

METHOD

This study employed a design study and targeted experimental approach guided by a Quality Management System (QMS) framework. The overall methodology followed the Plan-Do-Check-Act (PDCA) cycle, in which the planning stage defined quality objectives and selected appropriate model architectures and optimization strategies, the implementation stage deployed optimized models on Android devices, the evaluation stage applied standardized performance measurements, and the improvement stage refined optimization parameters based on the evaluation results.

The experimental tasks focused on two representative on-device AI applications: human presence detection and keyword spotting. The Visual Wake Words dataset Chowdhery et al.,

(2019) was used for the human presence detection task, whereas the Speech Commands dataset Warden, (2018) was used for the keyword spotting task, consistent with lightweight approaches developed for small-footprint keyword spotting on TinyML platforms (Garai & Samui, 2026).

Lightweight AI models suitable for mobile deployment were selected and optimized using complementary techniques, including 8-bit post-training quantization and structured channel pruning (Ghimire, Lee, & Kim, 2023; Wang & Zhu, 2020). Adaptive inference mechanisms based on early-exit architectures were also incorporated to reduce computational overhead while maintaining inference accuracy.

The optimized models were deployed on Android devices and evaluated under standardized experimental conditions. Performance measurements included energy consumption per inference, average power consumption, inference latency, model accuracy, model size after optimization, peak memory usage, and the energy-delay product (EDP). These metrics were analyzed to evaluate the effectiveness of the proposed QMS framework in optimizing the energy efficiency of on-device AI inference.

RESULTS AND DISCUSSION

Qualitative experiments conducted on two tasks—human-presence detection on low-resolution images and short-duration keyword spotting—show that eight-bit quantization executed with hardware acceleration reduces energy per inference compared to full-precision baseline execution. This reduction occurs without accuracy degradation that would impair application functionality under controlled test scenarios, consistent with findings reported in the model optimization literature for similar compression ratios and model architectures (Palakonda, Tursunboev, Kang, & Moon, 2025).

Applying early-exit at moderate network depths also accelerates execution time and reduces the energy-delay product for relatively easy samples, while stricter confidence thresholds help maintain prediction quality for more difficult cases (Passalis, Raitoharju, Tefas, & Gabbouj, 2020). The proportion of inputs exiting at each depth varies between tasks: the vision task shows a higher proportion of early exits on images with clear foreground subjects, while the audio task shows more variable exit distributions depending on acoustic clarity and background noise conditions.

Adaptive policies that incorporate battery level and thermal signals stabilize tail latency when the device begins heating, keeping response times controlled under repeated test loads. When thermal throttling is detected—indicated by a sustained increase in tail latency beyond a defined threshold—the adaptation policy shifts execution to pathways that generate less heat, accepting a modest increase in average latency in exchange for more consistent performance. This trade-off is quantified using the energy-delay product metric, which captures the combined impact on both energy efficiency and user-perceived responsiveness.

Quantitative results—including energy per inference, average power, median and tail latency, test accuracy, post-compression model size, peak memory usage, and energy-delay product—are to be populated upon completion of measurements on two device classes representing different tiers of Android hardware capability. The measurement protocol standardizes all experimental conditions as described, and all artifacts and analysis scripts have been prepared to support consistent reporting. This structure ensures that when quantitative data

are collected, they can be directly inserted into the result tables with full traceability to the documented protocol.

The findings align with theoretical explanations that computational load and memory traffic decrease when the precision of weights and activations is reduced. On low-power Android devices, performance bottlenecks often arise from memory access and cache efficiency rather than raw arithmetic throughput. Therefore, compression that reduces data footprint not only speeds up arithmetic operations but also lowers energy costs from repeated data movement between memory hierarchies. The fact that benefits appear in both vision and audio tasks suggests that the gains arise from on-device execution characteristics rather than dataset-specific quirks, supporting the generalizability of the QMS-guided approach across application domains.

Early-exit mechanisms act as adaptive valves that enable savings on easy samples without reducing overall reliability. When thresholds are chosen conservatively, the system exits early only on clearly identifiable inputs, preventing significant accuracy loss. Conversely, overly loose thresholds increase energy savings but also classification risk. A practical approach is to conduct short per-device calibration using trade-off curves between the energy-delay product and accuracy on a validation subset (Pacheco, Couto, & Simeone, 2023). This positions the threshold in a region that preserves predictive quality while yielding meaningful energy savings, and the calibration procedure itself becomes a documented step in the PDCA cycle's Act phase.

Policy controllers monitoring battery level and temperature help maintain service quality as device conditions degrade. When the device heats up or battery level drops, performance degradation often manifests as elongated tail latency due to thermal throttling of the CPU or accelerator. By switching execution pathways or raising early-exit probability when energy becomes a priority, the system sustains more stable latency profiles. This dynamic adaptation is particularly important for long-running applications such as continuous activity monitoring or always-on keyword detection, where device thermal and battery conditions change significantly over the application session.

Within a Quality Management System, the Plan-Do-Check-Act cycle provides a structured pathway linking technical decisions with measurement evidence, consistent with evidence on the effectiveness of PDCA-based quality improvement in other technical and clinical domains (Tamher, Rachmawaty, & Erika, 2021). Planning translates quality goals—such as maximum acceptable latency regression or minimum required accuracy—into operational metrics and defines the experimental scenarios needed to evaluate them. Execution applies quantization and structured pruning under documented procedures, ensuring that compression steps are reproducible. Checking standardizes measurement protocols so that results are comparable across experiments and devices. Acting adjusts thresholds, execution pathways, and compression parameters based on deviations from targets, with each adjustment documented and justified by evidence. This systematic approach transforms AI optimization from an ad-hoc process into an auditable engineering discipline.

The heterogeneity of Android hardware and software is a significant threat to external validity, a challenge also reported for heterogeneous edge–cloud DNN deployments (Yang, Shen, Zhong, & Liao, 2022). Differences in accelerator support, runtime library versions, and CPU scheduling policies may substantially alter energy and latency profiles even for the same

model and the same nominal device specification. Mitigations include replication across device classes, strict environmental control during measurements, and cross-validation using precision power supplies when available. Because absolute figures may still vary across devices, reporting emphasizes patterns and relative changes—such as the percentage reduction in energy per inference from quantization—that remain more robust across platforms than absolute values.

Ethical and privacy considerations are also important dimensions of on-device AI deployment. On-device processing generally reduces the movement of sensitive data, lowering the risk of data leakage during transmission, an advantage similarly emphasized in privacy-preserving edge federated learning frameworks (Aminifar, Shokri, & Aminifar, 2024). However, early-exit adaptation may treat certain data subsets unevenly if confidence thresholds are not well calibrated across demographic or environmental subgroups. Monitoring performance on relevant subsets—such as accent variation for audio or lighting conditions for vision—should therefore be included in the quality checklist as a fairness and robustness requirement. Periodic testing against distribution shifts is also recommended to ensure that optimized models maintain their performance as real-world input distributions evolve.

The practical implications for university laboratories are significant. With a modular artifact repository—comprising base models, compression recipes, test applications, and analysis scripts—students can rerun experiments on their own devices and populate result sections with their own measurements. This approach encourages healthy experimental literacy: instead of copying numbers from published papers, each group builds its own tables and charts and must explain why their results differ from those of other groups, developing critical thinking skills alongside technical competence. The QMS framework further reinforces this pedagogical value by requiring students to document their experimental procedures, justify their design choices, and propose evidence-based improvements.

CONCLUSION

This study demonstrates that a Quality Management System (QMS)-based framework can effectively guide technical decision-making to achieve energy-efficient on-device AI inference without compromising the user experience on resource-constrained Android devices. The combination of lightweight model architectures, 8-bit post-training quantization, calibrated early-exit mechanisms, adaptive device-state policies, and standardized measurement protocols provides a robust and reproducible foundation for continual improvement in on-device AI performance.

By monitoring quality indicators such as energy consumption per inference, median and tail inference latency, model accuracy, compressed model size, and the energy-delay product (EDP) within the Plan-Do-Check-Act (PDCA) cycle, organizations and research groups can systematically identify deviations from quality objectives, implement corrective actions, and objectively evaluate their effectiveness. The modular system architecture enhances traceability by enabling performance anomalies to be associated with specific system components and addressed without disrupting the overall implementation. Consistent findings across both vision and audio tasks indicate that improvements in energy efficiency primarily resulted from more efficient interactions between AI models and memory resources, suggesting that the proposed framework is applicable across diverse workloads and Android device classes.

Future research should validate these findings through comprehensive quantitative evaluations across a broader range of Android devices. Priority areas include knowledge distillation to preserve or improve model accuracy after compression, enhanced uncertainty estimation techniques to support more selective early-exit decisions, and context-aware adaptation policies, such as energy-saving operation under low-battery conditions and accuracy-prioritized operation when external power is available. The integration of on-device federated learning represents another promising direction, enabling continual model improvement while preserving data privacy through decentralized learning (Xia, Ye, Tao, Wu, & Li, 2021; Akhtarshenas, Vahedifar, Ayoobi, Maham, & Alizadeh, 2024). In higher education environments, the research artifacts developed in this study can support experimental replication and quality auditing, thereby strengthening experimental rigor and preparing students for quality-oriented engineering practice, in line with recent calls for standardized, reproducible evaluation practices across the tiny machine learning field (Velasquez, Cadavid, & Franco, 2025). Reinforced by the Plan-Do-Check-Act (PDCA) cycle, the proposed framework provides a practical, reproducible, and transferable roadmap for integrating on-device AI engineering with systematic quality management principles.

REFERENCES

Akhtarshenas, A., Vahedifar, M. A., Ayoobi, N., Maham, B., & Alizadeh, T. (2024). Federated learning: A cutting-edge survey of the latest advancements and applications. *Computer Networks*. https://doi.org/10.1016/j.comnet.2024.110752

Aminifar, A., Shokri, M., & Aminifar, A. (2024). Privacy-preserving edge federated learning for intelligent mobile-health systems. *Future Generation Computer Systems, 161*, 625–637. https://doi.org/10.1016/j.future.2024.07.032

Chowdhery, A., Warden, P., & Howard, A. (2019). *Visual Wake Words: On-device person detection dataset*. arXiv. https://arxiv.org/abs/1906.05721

Garai, S., & Samui, S. (2026). Advances in small-footprint keyword spotting for TinyML: A comprehensive review of efficient models and algorithms. *Neurocomputing, 695*, 134028. https://doi.org/10.1016/j.neucom.2026.134028

Ghimire, D., Lee, K., & Kim, S. (2023). Loss-aware automatic selection of structured pruning criteria for deep neural network acceleration. *Image and Vision Computing, 136*, 104745. https://doi.org/10.1016/j.imavis.2023.104745

Marinó, G. C., Petrini, A., Malchiodi, D., & Frasca, M. (2023). Deep neural networks compression: A comparative survey and choice recommendations. *Neurocomputing, 520*, 152–170. https://doi.org/10.1016/j.neucom.2022.11.072

Mi, J.-X., Feng, J., & Huang, K.-Y. (2022). Designing efficient convolutional neural network structure: A survey. *Neurocomputing, 489*, 139–156. https://doi.org/10.1016/j.neucom.2021.11.086

Pacheco, R. G., Couto, R. S., & Simeone, O. (2023). On the impact of deep neural network calibration on adaptive edge offloading for image classification. *Journal of Network and Computer Applications, 217*, 103684. https://doi.org/10.1016/j.jnca.2023.103684

Palakonda, V., Tursunboev, J., Kang, J.-M., & Moon, S. (2025). Metaheuristics for pruning convolutional neural networks: A comparative study. *Expert Systems with Applications, 268*, 126326. https://doi.org/10.1016/j.eswa.2024.126326

Passalis, N., Raitoharju, J., Tefas, A., & Gabbouj, M. (2020). Efficient adaptive inference for deep convolutional neural networks using hierarchical early exits. *Pattern Recognition, 105*, 107346. https://doi.org/10.1016/j.patcog.2020.107346

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Tamher, S. D., Rachmawaty, R., & Erika, K. A. (2021). The effectiveness of Plan Do Check Act (PDCA) method implementation in improving nursing care quality: A systematic review. *Enfermería Clínica, 31*, S54–S58. https://doi.org/10.1016/j.enfcli.2021.07.006

Tan, M., & Le, Q. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. *Proceedings of the 36th International Conference on Machine Learning*.

Velasquez, J. D., Cadavid, L., & Franco, C. J. (2025). Emerging trends and strategic opportunities in tiny machine learning: A comprehensive thematic analysis. *Neurocomputing, 648*, 130746. https://doi.org/10.1016/j.neucom.2025.130746

Wang, W., & Zhu, L. (2020). Structured feature sparsity training for convolutional neural network compression. *Journal of Visual Communication and Image Representation, 71*, 102867. https://doi.org/10.1016/j.jvcir.2020.102867

Warden, P. (2018). *Speech Commands: A dataset for limited-vocabulary keyword spotting*. arXiv. https://arxiv.org/abs/1804.03209

Xia, Q., Ye, W., Tao, Z., Wu, J., & Li, Q. (2021). A survey of federated learning for edge computing: Research problems and solutions. *High-Confidence Computing, 1*(1), 100008. https://doi.org/10.1016/j.hcc.2021.100008

Yang, L., Shen, X., Zhong, C., & Liao, Y. (2022). On-demand inference acceleration for directed acyclic graph neural networks over edge-cloud collaboration. *Journal of Parallel and Distributed Computing, 169*, 138–153. https://doi.org/10.1016/j.jpdc.2022.06.012