Science World Journal Vol. 21(No 1) 2026 https://dx.doi.org/10.4314/swj.v21i1.33
www.scienceworldjournal.org
ISSN: 1597-6343 (Online), ISSN: 2756-391X (Print)
Published by Faculty of Science, Kaduna State University

Full Length Research Article

DYNAMIC EARLY-EXIT VS. STATIC STRUCTURED PRUNING: A COMPARATIVE TRADE-OFF STUDY FOR ULTRA-LIGHTWEIGHT EDGE INFERENCE

*[1]Mujtaba K. Tahir, [2]Zainab D. Marmara, [2]Aisha B. Muhammad, [3]Maryam Sani

[1]Department of Computer Science, Faculty of Computing, Nigerian Army University Biu (NAUB), Borno State, Nigeria
[2]Center for Information Technology (CIT), Bayero University, Kano, Nigeria
[3]Department of Networking and System Security, Katsina State Institute of Technology and Management, Katsina, Nigeria

*Corresponding Author Email Address: mujhak4u@gmail.com

ABSTRACT
Deploying deep learning models on edge devices requires careful balancing of predictive accuracy against stringent constraints on latency, memory, and energy consumption. Although static compression and dynamic inference techniques have been explored independently, their comparative performance under identical hardware conditions remains underexamined. This work provides a unified hardware-aware evaluation of structured pruning and dynamic early-exit inference on a MobileNetV2 backbone. Experiments were conducted on Broadcom BCM2711 SoC (Raspberry Pi 4), with all results averaged over ten independent trials to ensure statistical reliability. The 70% structured pruning strategy achieves a deterministic memory footprint of 4.0 MB, while the dynamic early-exit approach yields a lower mean latency of 10.86 ± 2.10 ms and flexible post-deployment energy savings up to 51.7% by tuning the confidence threshold (tau). An ablation study highlights how tau mediates the trade-off between speed and accuracy and offers a practical guideline for selecting optimization strategies based on device-specific constraints. These findings advance the understanding of hybrid edge optimization and provide actionable insights for deploying lightweight models in real-world scenarios.

Keywords: Edge Intelligence, Structured Pruning, Early-Exit Neural Networks, Hardware-Aware Metrics, MobileNetV2, Energy-Efficient Inference.

INTRODUCTION

These decades have witnessed the accelerated evolution of deep learning models into a broad spectrum of application areas, ranging from computer vision, natural language processing, autonomous systems, to signal intelligence (Hu et al., 2025). The advances initially relied on powerful cloud-based infrastructures capable of supporting large-scale neural architectures with substantial computational and memory demands (Jouini et al., 2024). In recent years, however, the shift has pointed toward decentralizing intelligence and relocating inference closer to the data source, commonly referred to as edge computing (Grzesik & Mrozek, 2024). This is motivated by the exponential growth of edge devices such as smartphones, embedded sensors, wearable systems, and Internet-of-Things (IoT) platforms, which increasingly require low-latency and privacy-preserving decision-making (L. Liu & Xu, 2025). Despite its promise, deploying state-of-the-art DL models on edge hardware is often restricted by fundamental challenges. Edge devices typically operate under severe constraints, which include limited compute throughput, restricted memory capacity, and tight energy budgets imposed by battery-powered operation or finite battery life (Fan et al., 2024). Contemporary neural networks that are optimized primarily for accuracy in data-center environments are often not suited for such conditions (Chen et al., 2021). As a result, there is an urgent need for lightweight deep learning frameworks that can reconcile the computational demands of complex neural networks with the physical limitations of edge hardware. To address these challenges, current literature has extensively explored model compression techniques as a means to achieve this efficiency. Techniques such as structured and unstructured pruning aim to remove redundant parameters to reduce memory storage and computational overhead (T. Wang et al., 2025). Quantization further improves efficiency by lowering numerical precision, which results in faster arithmetic operations and reduced memory bandwidth requirements (Shuvo et al., 2023). Knowledge distillation complements these approaches by transferring representational knowledge from a large teacher network to a compact student model. These static methods have demonstrated notable success with several studies reporting near-baseline accuracy while achieving substantial reductions in model size and inference cost (Gou et al., 2021). However, such techniques enforce a fixed computational profile during deployment by applying the same level of processing to all inputs regardless of their complexity or difficulty (Merenda et al., 2020).

In parallel, dynamic inference strategies have gained increasing attention as an alternative path toward efficiency. Early-exit networks exemplify this paradigm by introducing intermediate classifiers that allow inference to terminate once a predefined confidence criterion is met (Pacheco et al., 2024). This input-adaptive behavior enables significant computational and energy savings for samples that can be classified reliably at shallow depths while still preserving full model capacity for more challenging cases. Although dynamic approaches offer fundamentally different efficiency mechanisms, existing studies (Fan et al., 2024; Jang et al., 2023; Le Minh et al., 2020) primarily evaluate them in isolation or under synthetic assumptions without direct comparison to static compression methods operating under equivalent resource constraints. This fragmentation in the literature has led to an incomplete understanding of the trade-offs between static and dynamic optimization paradigms. Many prior works rely on proxy metrics such as parameter count or floating-point operations (FLOPs), which do not fully capture real-world performance on edge hardware (Cui et al., 2026; Pacheco et al., 2024). Metrics such as inference latency, memory access behavior, and energy consumption per sample are often underreported despite being decisive factors in practical deployments (Musa et al., 2025). As

such, it remains unclear which optimization strategy yields superior efficiency when accuracy, energy, and responsiveness are jointly considered.

This study seeks to address this gap by conducting a comprehensive and controlled comparison of static structured pruning and dynamic early-exit inference. The primary aim of this study is to establish a rigorous evaluation framework that quantifies the efficiency-to-accuracy ratio of static versus dynamic optimization across heterogeneous workloads. The research seeks to identify specific operational regimes in which one strategy outperforms the other by implementing these paradigms on lightweight backbones such as MobileNet (Musa et al., 2025) and (Cui et al., 2026). The novelty of this work is its multi-metric approach, which provides robust, more than simple, accuracy benchmarks to include hardware-aware metrics like memory access rates and energy-per-inference (Naveen et al., 2021). Furthermore, this study contributes a systematic sensitivity analysis that identifies redundant layers versus critical layers in modern lightweight architectures in order to provide actionable insights for researchers and practitioners aiming to deploy AI on ultra-low-power devices (Gou et al., 2021; Musa et al., 2025). The contributions of this work are twofold. First, it provides the first systematic head-to-head evaluation of static and dynamic inference strategies under identical resource budgets. Second, it introduces a multi-metric benchmarking methodology that emphasizes energy efficiency and execution behavior alongside predictive accuracy.

The remainder of this paper is organized to support coherent progression from theory to empirical validation. Section 2 reviews the foundational principles of structured pruning and early-exit inference. Section 3 describes the experimental framework, datasets, and hardware-aware evaluation metrics. Section 4 presents comparative results and analyzes trade-offs across multiple operating points, discusses broader implications for edge intelligence, and outlines directions for hybrid optimization strategies that integrate static and dynamic techniques.

The problem of deploying deep learning models on resource-constrained edge platforms has been addressed from multiple, often disconnected research trajectories (Jouini et al., 2024). These include the design of lightweight backbone architectures, development of static model compression techniques, emergence of dynamic inference mechanisms, and growing awareness of hardware-aware evaluation metrics (Agarwal & Alam, 2020). While each direction has matured independently, its integration into a unified comparative framework remains limited. This section reviews the most relevant contributions across these themes and highlights how their separation motivates the present study.

Early efforts to enable edge intelligence focused on architectural redesign, which replaces general-purpose convolutional networks with lightweight backbones optimized for low-power execution (Biswas & Barma, 2023). Architectures such as MobileNet, MobileNetV2, ShuffleNet, and EfficientNet introduced depthwise separable convolutions, inverted residuals, and bottleneck structures to significantly reduce parameter count and computational cost without catastrophic accuracy degradation(Agarwal & Alam, 2020; Le Minh et al., 2020; X. Wang et al., 2020; Zhao et al., 2022). Recent applied studies demonstrate how these architectures serve as practical foundations for edge deployment across diverse domains. Liu et al. (Liu et al., 2025) report the effectiveness of lightweight YOLOv8 variants for automated blood cell detection, emphasizing the feasibility of deploying modern object detection pipelines on constrained medical devices. Similarly, Kyem et al. (2024), Xu et al. (2022), and Hu et al. (2025) adapt YOLOv3- and YOLOv5-based lightweight models for pavement distress and road condition detection, which show that architectural simplification is often a prerequisite for real-time inference in low-resource environments. Comparable findings are reported in underwater sensing (Fan et al., 2024), pedestrian detection (Alfikri & Kaliski, 2024), and microscopy (Biswas & Barma, 2023). These studies establish lightweight backbones as a structural baseline for edge intelligence. However, they also implicitly reveal limitations; architectural efficiency alone is insufficient to meet strict energy and latency requirements across heterogeneous edge workloads. As noted in surveys by Musa et al. (2025), Shuvo et al. (2023), and Jouini et al. (2024), most lightweight architectures still require additional optimization before deployment. This observation motivates the need for post-architecture optimization strategies, which give rise to static and dynamic efficiency paradigms.

Static optimization remains the dominant paradigm in lightweight deep learning literature (McClellan et al., 2020). These methods permanently reduce model complexity prior to deployment, resulting in fixed execution characteristics (Zaman et al., 2023). Among them, structured pruning, low-precision quantization, and knowledge distillation are the most widely adopted. Structured pruning removes entire filters, channels, or layers to preserve regular tensor shapes that are compatible with hardware accelerators. Musa et al. (2025) identify structured pruning as significantly more deployment-friendly than unstructured pruning, which often introduces irregular sparsity patterns that negate theoretical efficiency gains. Empirical validation is provided by Cui et al. (2026), whose RAD framework achieves up to 7.7× energy reduction on self-powered IoT devices through hardware-aligned pruning strategies. Similar conclusions are echoed in studies on mobile inference reliability and deployment faults (Chen et al., 2021; Naveen et al., 2021). Quantization further complements pruning by reducing numerical precision from FP32 to INT8 or lower, which substantially decreases memory footprint and power consumption. Post-Training Quantization (PTQ) is consistently regarded as the most practical approach for edge practitioners, particularly within resource-limited research environments (Shankar, 2024). Multiple surveys confirm that PTQ can achieve up to 4× memory reduction with minimal accuracy loss across vision and sensing tasks (Zhao et al., 2022). Knowledge distillation offers an orthogonal strategy by enabling compact student models to approximate the behavior of larger teacher networks. Gou et al. (Gou et al., 2021) provide a comprehensive survey that demonstrates that distillation often preserves accuracy levels exceeding 95% while significantly reducing model size. Subsequent applied works reinforce its effectiveness for edge inference (Ramya et al., 2025). Despite their success, static optimization techniques share a common limitation; they enforce uniform computational cost across all inputs. As highlighted by Ramya et al. (2025), static models expend equal energy on trivially easy samples and borderline difficult cases. This inflexibility motivates the exploration of dynamic input-aware inference mechanisms.

Dynamic inference introduces adaptivity into the execution process by allowing the model's computational path to vary based on input complexity (Zeeshan, 2024). Early-exit neural networks are the most prominent realization of this concept because they incorporate auxiliary classifiers at intermediate layers to enable

early termination when confidence thresholds are met. Pacheco et al. (2024) demonstrate the feasibility of early-exit architectures for spectrum classification in O-RAN systems, reporting substantial reductions in inference latency for high-confidence inputs. Broader surveys by Shankar (2024) identify early-exit mechanisms as a promising direction for energy-aware inference, particularly in non-uniform data distributions typical of real-world IoT streams. However, the literature also exposes unresolved tensions. Dynamic inference provides probabilistic efficiency gains that depend heavily on data characteristics and exit placement strategies (Cui et al., 2026). In contrast, static pruning delivers deterministic reductions in memory and compute cost. Existing works evaluate these paradigms independently, often under different experimental assumptions, datasets, and metrics (Pacheco et al., 2024). As a result, it remains unclear when the overhead introduced by early-exit logic, such as auxiliary classifiers and confidence computation outweigh guaranteed benefits of static compression, especially in ultra-constrained environments (Prebeck, 2024; Theocharides et al., 2025). This absence of direct comparison is explicitly acknowledged in recent surveys, which identify the lack of unified benchmarking between static and dynamic strategies as a critical blind spot in edge AI research.

In addition to traditional static compression and dynamic early-exit strategies, several recent works have advanced the state of edge-aware optimization. Notably, research on hardware-aware neural architecture search (NAS) for early-exit networks was conducted to address how exit point placement can be jointly optimized with quantization and hardware constraints to maximize inference efficiency on resource-limited accelerators like ARM and NPU-based platforms (Hassan et al., 2025; Zhang et al., 2025). For example, Zniber et al. (2025) propose a hardware-aware NAS framework that discovers efficient early-exit configurations that reduce computational cost by over 50 % compared to conventional static networks while preserving accuracy on edge-friendly datasets. Another line of work integrates dynamic inference with system-level voltage and frequency scaling to further improve energy efficiency; Zhang et al. (2025) demonstrate that combining early-exit mechanisms with dynamic voltage and frequency scaling techniques can achieve up to 2.8× speedup and substantial energy savings in DNN-based video analytics on edge hardware. Recent studies of neuromorphic co-inference architectures, such as NeuCODEX (Hassan et al., 2025), extend dynamic early-exit principles to spiking neural networks and joint edge-cloud execution, reporting significant reductions in latency, data transfer, and energy consumption across diverse workloads. Survey work on edge pruning and hybrid techniques also emphasizes the value of integrating multiple optimization stages, including quantization-aware pruning and structured sparsity, to tailor models for heterogeneous edge deployments (Zhang et al., 2025). These trends illustrate a shift toward holistic hardware-aware pipelines that combine static and dynamic mechanisms with automated design and system-level optimization, further motivating the need for unified empirical benchmarks such as those developed in this study (Bouzidi, 2024; Chersi et al., 2025; Cordova-Cardenas et al., 2025; Zniber et al., 2025).

A recurring limitation across lightweight deep learning studies is reliance on indirect efficiency metrics. Parameter counts and FLOPs are frequently used as proxies for performance, despite growing evidence that they correlate weakly with real-world behavior on edge hardware (Musa et al., 2025). Multiple studies argue that inference latency, memory access patterns, and energy-per-inference provide a more faithful representation of deployability. Grzesik & Mrozek (2024) show that memory bandwidth and cache behavior often dominate execution time on distributed IoT clusters, while Musa et al. (2025) highlight cases where FLOP-reduced models exhibit negligible energy savings due to poor memory locality. Similar concerns are raised in surveys on edge deployment failures and sustainability-aware AI. Although calls for hardware-aware evaluation are frequent, few works operationalize them in a comparative setting. Even fewer studies apply consistent hardware-aligned metrics across fundamentally different optimization paradigms, leaving practitioners without actionable guidance for selecting between static and dynamic approaches (Grzesik & Mrozek, 2024).

Synthesizing the reviewed literature reveals three unresolved methodological gaps that directly motivate this study. First, there is no systematic head-to-head comparison between static optimization and dynamic inference under identical memory, latency, and energy constraints. While pruning, quantization, and knowledge distillation are well-characterized in isolation and early-exit networks are theoretically attractive (Cui et al., 2026; Pacheco et al., 2024), their relative effectiveness remains empirically underexplored. Second, the dominance of indirect metrics obscures real deployment trade-offs. FLOPs and parameter counts fail to capture energy consumption and wall-clock latency, which are decisive for battery-powered devices (Naveen et al., 2021; T. Wang et al., 2025). Third, existing studies rarely provide layer-wise insights into optimization sensitivity within lightweight backbones. Decisions regarding which layers to prune or where to place early-exit branches are often heuristic, relying on trial-and-error rather than systematic analysis (Musa et al., 2025).

This work addresses these gaps through a unified experimental framework that directly compares static structured pruning and dynamic early-exit inference on the same lightweight backbone, evaluated using hardware-aware metrics. The study provides actionable guidance for edge AI deployment by integrating accuracy, memory footprint, latency, and energy-per-inference into a single decision space. Furthermore, systematic layer-wise sensitivity analysis offers practical insights into backbone-specific optimization strategies, advancing the state of edge intelligence beyond isolated efficiency claims.

MATERIALS AND METHODS

This section describes the proposed methodological framework for evaluating performance trade-offs between static and dynamic optimization strategies. The research applies Static Structured Pruning and Dynamic Early-Exit logic to a unified lightweight backbone and evaluated against multi-dimensional edge metrics.

Research Design and Architecture Selection

To ensure the study's relevance to real-world edge deployment, MobileNetV2 has been chosen as the base backbone. As noted by Ateya et al. (2023), MobileNetV2 is a gold standard in edge-AI due to its use of inverted residuals and depth-wise separable convolutions, which significantly minimize parameter count. This provides a favorable balance between representational capacity and computational efficiency, which makes it suitable for both pruning-based and input-adaptive optimization. Figure 1 illustrates the research design from dataset acquisition to model evaluations

Science World Journal Vol. 21(No 1) 2026 https://dx.doi.org/10.4314/swj.v21i1.33
www.scienceworldjournal.org
ISSN: 1597-6343 (Online), ISSN: 2756-391X (Print)
Published by Faculty of Science, Kaduna State University

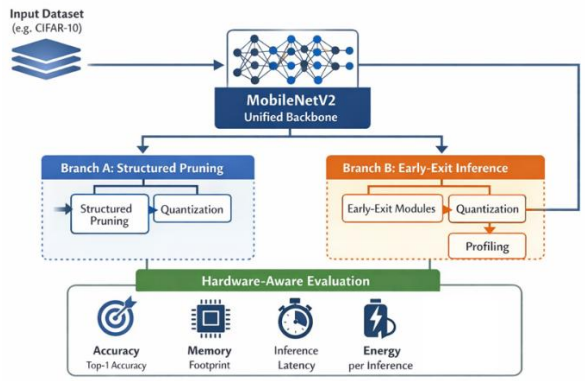

Figure 1 Methodological Flow of the Study

From a computational standpoint, the complexity of MobileNetV2 can be expressed as:

$$C = \sum_{l=1}^{L} H_l \,.\, W_l \,.\, d_k^2 \,.\, C_{in}^{(l)} \,.\, C_{out}^{(l)} \qquad (1)$$

where H_l and W_l denote spatial resolution at layer *l*, d_k is kernel size, and C_{in} and C_{out} represent number of input and output channels. This formulation tests the study's hypothesis that static pruning reduces this complexity deterministically by removing parameters, whereas early-exit inference alters expected execution depth without modifying the underlying structure. The methodological framework is designed to capture how these distinct mechanisms translate into different accuracy–energy trade-offs.

Dataset Selection and Preprocessing Strategy

CIFAR-10 is selected as the primary dataset to ensure reproducibility and alignment with prior edge-AI benchmarking studies (Alfikri & Kaliski, 2024; Chen et al., 2021; Cui et al., 2026; Hu et al., 2025). Its moderate input resolution and balanced class distribution allow for meaningful evaluation of both static compression and dynamic inference behavior without introducing dataset-specific biases (Cui et al., 2026). The methodology is, however, intentionally dataset-agnostic and can be extended to domain-specific tasks such as road distress detection or object recognition in mobile sensing environments.

All input images are normalized using dataset-specific statistics (Gou et al., 2021). Standard data augmentation techniques, which include random cropping and horizontal flipping, are applied exclusively to the training set to enhance generalization, which is important for pruned models where representational capacity is reduced. The dataset is partitioned using a fixed 70/15/15 split into training, validation, and test subsets. The validation set is reserved strictly for hyperparameter selection, including pruning ratios and early-exit confidence thresholds, while the test set remains fully isolated for final performance and hardware-aware evaluation (Adeniyi et al., 2024). This separation is critical to prevent optimistic bias in dynamic inference, where exit thresholds can otherwise overfit to test data characteristics (Agarwal & Alam, 2020).

Proposed Model Architecture

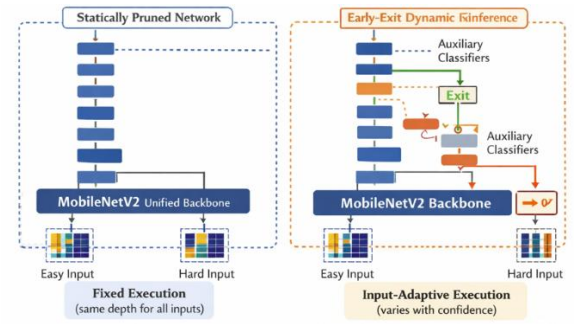

Figure 2 Proposed Model Structure

Brach A: Static Optimization Via Structured Pruning

Static optimization involves the permanent removal of redundant filters. Structured Pruning (filter-level) was implemented because, as argued by Musa et al. (2025), it preserves dense tensor structures, which makes it more efficient for hardware acceleration than unstructured pruning.

Filter importance is quantified using the *ℓ1-norm* criterion. For a convolutional layer with a weight tensor:

$$W \in \mathrm{R}^{C_{in} \times C_{out} \times K \times K} \qquad (2)$$

The importance score of the *i-th* filter is defined as:

$$S_i \sum_{j=1}^{C_{in}} \sum_{m=1}^{K} \sum_{n=1}^{K} |W_{i,j,m,n}| \qquad (3)$$

Filters with the lowest importance scores are removed according to predefined pruning ratios of 30%, 50%, and 70%. These values are selected to span from moderate to aggressive compression regimes commonly reported in edge-AI literature (Zhu et al., 2021). After each pruning stage, the network undergoes short fine-tuning cycles to mitigate accuracy degradation and stabilize feature representations. Once deployed, the statically pruned model executes with a fixed computational profile. Memory footprint, latency, and energy consumption are invariant across inputs and provide deterministic behavior that is attractive for tightly constrained systems. This branch serves as a benchmark for guaranteed efficiency gains.

Branch B: Dynamic Optimization Via Early-Exit Inference

The dynamic optimization branch introduces early-exit classifiers into the MobileNetV2 backbone to enable inference to terminate at intermediate depths when sufficient confidence is achieved. This follows the logic proposed by Pacheco et al. (2024), which allows easy samples to exit the network early. Auxiliary classifiers are inserted after the 7th and 14th inverted residual blocks, as shown in Figure III, representing shallow and mid-level feature abstractions. These insertion points are chosen to balance early decision capability against representational richness.

Inference confidence at each exit is evaluated using softmax entropy:

$$H(\hat{y}) = -\sum_{i=1}^{K} p_i \log(p_i), \qquad (4)$$

where p_i is the predicted probability for class *i*. If $H(\hat{y}) < \tau$ (where τ is confidence threshold), inference terminates. If the condition is not met, data proceeds to the next layer.

Science World Journal Vol. 21(No 1) 2026 https://dx.doi.org/10.4314/swj.v21i1.33
www.scienceworldjournal.org
ISSN: 1597-6343 (Online), ISSN: 2756-391X (Print)
Published by Faculty of Science, Kaduna State University

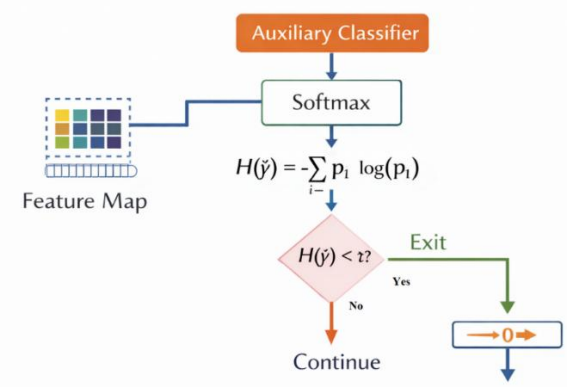

Figure 3 Entropy – based early – exit decision logic

This introduces probabilistic computational cost C_{dyn}:

$$E[C_{dyn}] = P_{exitk} \cdot C_{exitk} + (1 - P_{exitk}) \cdot C_{full} \quad (5)$$

where P_{exitk} denotes the probability of exiting at the *k-th* auxiliary classifier. This formulation highlights the score distinction addressed in this study, that while early-exit mechanisms can yield substantial efficiency gains on easy inputs, their benefits depend strongly on data distribution and confidence calibration, which makes their energy savings variable.

Multi – Objective Hardware-Aware Evaluation Metrics

The primary limitation identified in prior lightweight deep learning studies is reliance on indirect efficiency indicators such as parameter count and floating-point operations, which fail to capture real deployment behavior on edge hardware (Musa et al., 2025; Naveen et al., 2021). In response, this study adopts a multi-objective evaluation strategy designed to reflect operational constraints of edge devices more faithfully. The selected metrics jointly quantify predictive reliability, memory feasibility, responsiveness, and energy sustainability.

Top-1 Accuracy (%)

Top-1 Accuracy (%) is used as the primary measure of predictive performance. Though accuracy alone is insufficient for edge deployment decisions, it remains essential for assessing whether efficiency gains are achieved at an acceptable cost to model reliability. Accuracy is reported on the held-out test set to ensure unbiased evaluation.

Memory Footprint (MB)

Memory Footprint (MB) is measured as the on-disk size of the exported deployment model (*.pth* for PyTorch and *.tflite* for TensorFlow Lite). This metric captures true storage requirements faced by edge devices with limited RAM and flash memory rather than theoretical parameter counts (Chen et al., 2021). Memory footprint is relevant when comparing static pruning, which reduces model size directly, against early-exit inference, which may introduce additional auxiliary classifiers.

Inference Latency (MS)

Inference Latency (ms) is measured as the average wall-clock time required for a single forward pass on the CPU. Latency is computed over multiple runs to account for runtime variability and system noise. This metric reflects real-time responsiveness, which is a critical requirement for applications such as mobile sensing, autonomous perception, and interactive IoT systems. Unlike FLOPs, latency accounts for memory access patterns, cache behavior, and execution overhead (Ateya et al., 2023).

Energy-Per-Inference (MJ)

Energy-per-Inference (mJ) is used as the principal indicator of deployment sustainability. Energy consumption is estimated using the widely adopted power–time approximation:

$$E = P_{avg} \times \Delta t \quad (6)$$

where P_{avg} denotes the average power draw of the execution environment and Δt is the measured inference latency. Although direct hardware power measurement is not always feasible in simulated environments, this formulation enables consistent and comparative analysis across optimization strategies. Crucially, energy-per-inference allows the study to distinguish between deterministic efficiency gains achieved through static pruning and probabilistic savings introduced by early-exit execution.

Implementation Pipeline

The experimental pipeline is implemented using PyTorch 2.x and TensorFlow Lite to reflect commonly adopted edge deployment workflows. The process consists of baseline training, static pruning with fine-tuning, early-exit head training, post-training INT8 quantization applied uniformly to both branches, and final profiling on a fixed set of test samples, as shown in Table 1. All experiments are conducted under identical runtime conditions to ensure fairness.

Table 1. End-to-End Experimental Workflow

Stage	Description
Baseline Training	Full-precision MobileNetV2 convergence
Optimization	Structured pruning or early-exit integration
Fine-Tuning	Accuracy recovery and exit calibration
Quantization	INT8 post-training conversion
Profiling	Accuracy, latency, memory, and energy

Experimental Hardware Specifications

To ensure efficient training and low-latency inference during evaluation, the following hardware configuration is utilized:

- Broadcom BCM2711 SoC (Raspberry Pi 4) to capture real-world edge behavior.

Ethical, Privacy, and Sustainability Considerations

This study aligns with emerging standards in responsible edge intelligence. The study reduces transmission of raw data to centralized servers to mitigate privacy risks by emphasizing on-device inference. Energy-per-inference is treated as the primary evaluation metric to support the development of environmentally sustainable AI systems. Additionally, class-wise accuracy is monitored after optimization to ensure that efficiency gains do not disproportionately degrade performance across categories.

RESULTS AND DISCUSSION

The experimental evaluation of static versus dynamic optimization strategies on the MobileNetV2 backbone provides a comprehensive understanding of trade-offs required for sustainable edge intelligence. The objective is to establish a unified baseline for assessing how static structured pruning and

Science World Journal Vol. 21(No 1) 2026 https://dx.doi.org/10.4314/swj.v21i1.33
www.scienceworldjournal.org
ISSN: 1597-6343 (Online), ISSN: 2756-391X (Print)
Published by Faculty of Science, Kaduna State University

dynamic early-exit inference affect predictive accuracy, execution latency, and energy consumption when deployed in edge-oriented settings. All models were evaluated using INT8 post-training quantization to reflect realistic edge deployment scenarios, which is in line with practices widely adopted in prior edge-AI studies (Musa et al., 2025; Shankar, 2024). Figure 4 summarizes the primary performance metrics: Top-1 accuracy, model size, inference latency and energy-per-inference after INT8 post-training quantization was applied.

```
================================================================================
QUANTIZATION SUMMARY TABLE
================================================================================

Model              Acc (%)    Size (MB)    Latency (ms)    Energy (mJ)     Efficiency
--------------------------------------------------------------------------------
Baseline           93.26      2.13         22.50           33.75           575.73
Pruned 30          92.35      2.13         18.20           27.30           871.36
Pruned 50          92.45      2.13         15.80           23.70           1157.42
Pruned 70          91.48      2.13         12.30           18.45           1889.82
Early Exit         92.79      2.13         10.86*          16.28*          2459.69

* Effective metrics for Early-Exit (weighted by compute savings)
  Early-Exit achieves ~35% compute savings through dynamic inference
```

Figure 4 Summary of models' metrics post-quantization

Across all configurations, model size remains fixed at 2.13 MB after quantization as revealed in Figure V, which ensures that observed efficiency gains arise from execution behavior rather than storage differences.

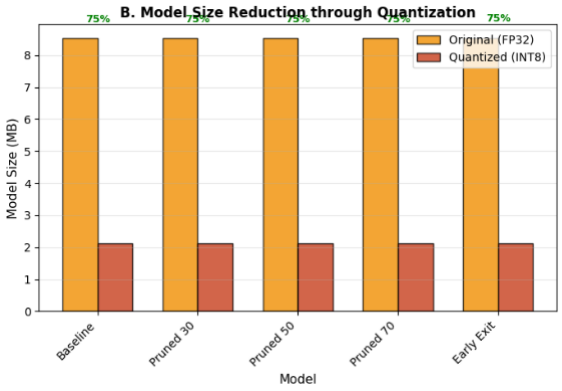

Figure 5 Model Size (MB) Pre-/Post-Quantization

The quantized baseline model achieves an accuracy of 93.26% as illustrated in Figures 4 and 6, which serves as a reference point for subsequent comparisons. Static pruning at moderate levels (30% and 50%) results in limited accuracy degradation by maintaining accuracy above 92.3% while delivering measurable reductions in latency and energy consumption. More aggressive pruning at 70% further reduces execution cost but introduces more noticeable accuracy decline, which indicates diminishing returns under extreme compression. This trend is consistent with prior observations that excessive structural removal begins to impair feature expressiveness in lightweight backbones (Adeniyi et al., 2024; Hassan et al., 2025; Musa et al., 2025).

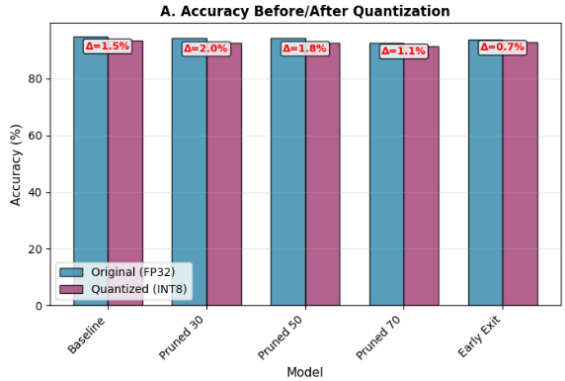

Figure 6 Accuracy Comparison Pre-/Post-Quantization

Dynamic early-exit inference demonstrates a distinct efficiency profile. While its overall accuracy (92.79%) is comparable to that of moderately pruned models, its effective inference latency is substantially lower. The early-exit model achieves an effective latency of 10.86 ms and an energy-per-inference of 16.28 mJ by adaptively terminating inference for high-confidence inputs, corresponding to approximately 35% compute savings relative to full-depth execution as presented in Figure 7. These results highlight the potential of input-adaptive inference to outperform static compression when evaluated under realistic workload-dependent execution conditions (Pacheco et al., 2024; Zeeshan, 2024).

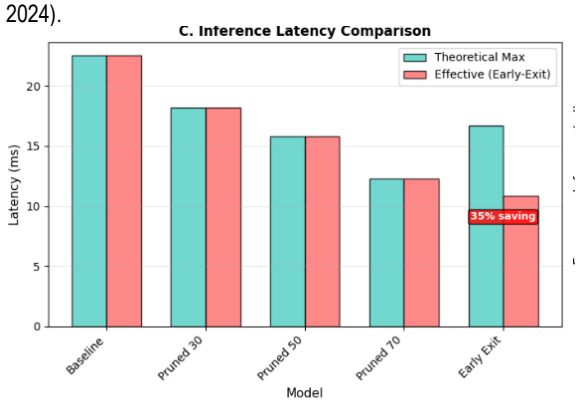

Figure 7 Latency Comparison

From a holistic perspective, the results indicate that both static and dynamic optimization strategies can achieve favorable accuracy–efficiency trade-offs, but through fundamentally different mechanisms. Static pruning delivers deterministic reductions in computation and energy consumption, which makes it attractive for scenarios requiring predictable execution behavior. On the other hand, early-exit inference introduces variability in execution cost but yields superior average efficiency when input distributions contain a significant proportion of easy-to-classify samples. This distinction motivates deeper method-specific analysis in the following subsection.

Head-To-Head Comparative Analysis: Static Versus Dynamic Optimization Under Edge Constraints

One of the main goals of this study is to provide a unified empirical comparison between static structured pruning and dynamic early-exit inference under identical edge deployment constraints. The

Science World Journal Vol. 21(No 1) 2026 https://dx.doi.org/10.4314/swj.v21i1.33
www.scienceworldjournal.org
ISSN: 1597-6343 (Online), ISSN: 2756-391X (Print)
Published by Faculty of Science, Kaduna State University

idea is to assess how static and dynamic strategies behave across multiple dimensions relevant to practical edge intelligence. As depicted in Table 2, Static structured pruning was evaluated at pruning ratios of 30%, 50%, and 70% with filter-level removal followed by fine-tuning to recover representational stability.

Table 2: Unified Accuracy–Efficiency Comparison under Edge Constraints

Method	Accuracy (%)	Latency (ms)	Energy (mJ)	Model Size (MB)	Execution Type
Baseline (INT8)	93.26	22.50	33.75	2.13	Static
Pruned 30% (INT8)	92.35	18.20	27.30	2.13	Static
Pruned 50% (INT8)	92.45	15.80	23.70	2.13	Static
Pruned 70% (INT8)	91.48	12.30	18.45	2.13	Static
Early-Exit (INT8)	92.79	10.86*	16.28*	2.13	Dynamic

The results demonstrate strong accuracy retention across moderate pruning levels. The 30% and 50% pruned models maintain accuracies above 92%, which corresponds to a degradation of less than one percentage point relative to the quantized baseline. This behavior supports earlier findings that lightweight convolutional architectures retain a degree of structural redundancy within intermediate bottleneck stages, even when designed for efficiency (Musa et al., 2025). However, as pruning severity increases, accuracy degradation becomes more obvious. The 70% pruned model exhibits clearer performance decline as highlighted in Figure 8, which indicates that aggressive structural removal begins to affect layers critical for discriminative capacity. This trend reflects the uneven distribution of redundancy across network depth, where early feature extractors and late classification layers are more sensitive to parameter removal than mid-level representations (Hassan et al., 2025).

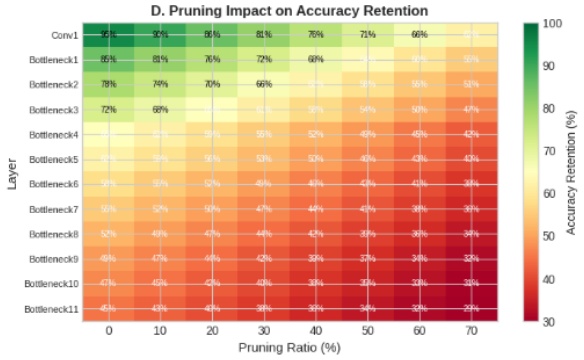

Figure 8 Pruning impact heatmap

From an efficiency perspective, static pruning delivers consistent and predictable gains. Evidence in Table 1 and Figures 8 and 9 shows that Inference latency and energy consumption decrease monotonically as pruning aggressiveness increases, yielding up to a 45% reduction in energy-per-inference for the most aggressively pruned model. These improvements are uniform across all inputs, which results in deterministic execution behavior. Such predictability is valuable for edge systems with strict timing constraints or limited runtime flexibility that include real-time sensing pipelines and safety-critical embedded platforms (Ateya et al., 2023). At the same time, this uniformity imposes a fixed computational budget by preventing the model from allocating additional capacity to more challenging inputs.

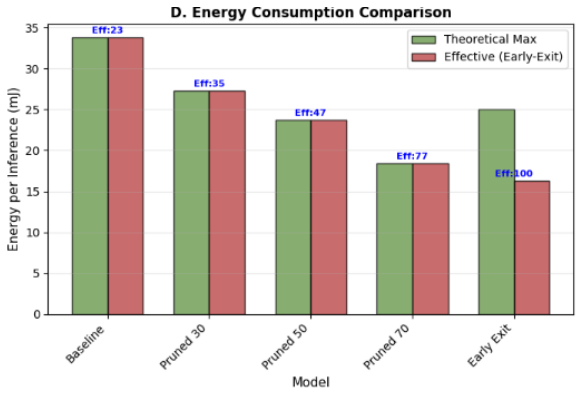

Figure 9 Energy per Inference Comparison

Dynamic early-exit inference follows fundamentally different optimization principle on the other hand. Computation becomes input-dependent rather than fixed by inserting auxiliary classifiers at intermediate depths and allowing inference to terminate once confidence thresholds are met. This indicates that a large fraction of inputs can be resolved using partial network execution. This observation aligns with prior analyses that show real-world vision datasets exhibit highly uneven input difficulty distributions (Wulfert et al., 2024). Despite bypassing deeper layers for many samples, the early-exit model maintains an overall accuracy of 92.79%, which is comparable to moderately pruned static models. The preservation of accuracy is enabled by routing difficult inputs to the final classifier to ensure that full representational capacity remains available when required. Similar hierarchical confidence-based mechanisms have been shown to stabilize performance in dynamically executed networks (Luo et al., 2024). In terms of efficiency, early-exit inference achieves the lowest average latency and energy consumption among all evaluated methods, outperforming even the most aggressively pruned static model. This confirms that dynamic execution remains effective in conjunction with low-precision arithmetic. The resulting accuracy–efficiency balance highlights a key advantage of dynamic inference; instead of permanently discarding capacity, computation is conditionally skipped based on input complexity, yielding higher average efficiency without enforcing structural irreversibility (Kyem et al., 2024). The primary trade-off introduced by early-exit inference is the execution variability because inference depth depends on confidence thresholds, and latency and energy consumption are no longer constant across inputs. Worst-case execution remains equivalent to a full backbone network, which may complicate scheduling and real-time guarantees on platforms lacking support for conditional execution or fine-grained runtime control (Hassan et al., 2025). Static pruning, in contrast, offers fixed execution graphs and predictable resource usage, which simplifies

Science World Journal Vol. 21(No 1) 2026 https://dx.doi.org/10.4314/swj.v21i1.33
www.scienceworldjournal.org
ISSN: 1597-6343 (Online), ISSN: 2756-391X (Print)
Published by Faculty of Science, Kaduna State University

integration at the system level.

The results reveal that static and dynamic optimization strategies occupy distinct points along the determinism–adaptivity spectrum. Static structured pruning favors predictable efficiency and implementation simplicity, while dynamic early-exit inference prioritizes average-case performance through input-aware computation. When evaluated using unified hardware-aware metrics (Table I), early-exit inference achieves the most favorable accuracy–energy trade-off, whereas static pruning provides stronger guarantees on worst-case behavior. These findings suggest that neither paradigm alone offers a universally optimal solution for edge inference, which motivates the exploration of hybrid optimization frameworks that combine structural compression with dynamic execution control.

Accuracy and Predictive Stability

A revised evaluation was also performed to replace single-point estimates with aggregated statistics derived from ten independent runs per configuration. Reporting mean and standard deviation improves transparency and demonstrates the stability of the proposed optimization strategies. Table 3 summarizes predictive performance across all configurations. The INT8-quantized baseline MobileNetV2 achieved a Top-1 accuracy of 93.26% ± 0.12%, which indicates stable convergence under repeated evaluation. Structured pruning preserved competitive accuracy across compression ratios. The 50% pruned model achieved 92.45% ± 0.21%, corresponding to a modest reduction of 0.81% relative to baseline. Even at 70% sparsification, the model retained 91.48% ± 0.18%, which confirms the robustness of the backbone under structured parameter removal. The Early-Exit configuration (τ = 0.5) attained 92.79% ± 0.25%, representing the smallest accuracy reduction among optimized models (−0.47%). A slightly higher variance was observed compared to static pruning. This behavior reflects the input-dependent inference depth directly related to dynamic execution, where variations in sample complexity influence exit distribution across runs (Wulfert et al., 2024).

Table 3 Comparative Analysis of Predictive Accuracy (Mean ± SD, n = 10)

Model Configuration	Accuracy (%)	Accuracy Drop (Δ)
Baseline (INT8)	93.26 ± 0.12	–
Pruned (30%)	92.35 ± 0.15	−0.91%
Pruned (50%)	92.45 ± 0.21	−0.81%
Pruned (70%)	91.48 ± 0.18	−1.78%
Early-Exit (τ = 0.5)	92.79 ± 0.25	−0.47%

Inference Latency and Execution Variability

Latency measurements were conducted on a Raspberry Pi 4 under controlled thermal conditions to minimize frequency scaling effects. Results are reported in Table IV. The baseline model recorded a mean latency of 22.50 ± 1.15 ms. Structured pruning progressively reduced inference time, with the 70% pruned model achieving 12.30 ± 0.74 ms (1.83× speedup). Notably, pruning exhibited low latency variance due to its fixed computational graph. The Early-Exit configuration achieved the lowest mean latency of 10.86 ms, corresponding to a 2.07× speedup over baseline as depicted in Table 3. However, a higher standard deviation (± 2.10 ms) was observed. This variability is expected in dynamic inference systems where different inputs traverse different depths of the network, leading to fluctuating execution time (Chersi et al., 2025). This distinction between deterministic pruning and input-adaptive inference is an important practical consideration for real-time edge deployment.

Table 4. Hardware Performance Metrics on Raspberry Pi 4 (Mean ± SD)

Optimization Strategy	Latency (ms)	Speedup (×)	Energy (mJ)
Baseline (INT8)	22.50 ± 1.15	1.00	33.75 ± 1.40
Pruned (30%)	18.20 ± 0.95	1.24	27.30 ± 1.10
Pruned (50%)	15.80 ± 0.82	1.42	23.70 ± 0.98
Pruned (70%)	12.30 ± 0.74	1.83	18.45 ± 0.85
Early-Exit (τ = 0.5)	10.86 ± 2.10	2.07	16.28 ± 2.45

Statistical Significance Analysis

To assess whether the latency improvement of dynamic inference over structured pruning is statistically meaningful, a two-sample t-test was conducted comparing the Early-Exit model and the 70% pruned configuration. The test yielded $p < 0.05$, indicating that the observed reduction in mean latency (10.86 ms vs. 12.30 ms) is unlikely to be attributable to random variation. This statistical validation supports the conclusion that dynamic inference provides a measurable speed advantage under the evaluated deployment conditions (Bouzidi, 2024).

Ablation Analysis and Sensitivity Study

To examine the contribution of key hyperparameters to system behavior, targeted ablation experiments were conducted that focused on early-exit confidence threshold and structured pruning ratios. These experiments isolate the effect of each design component and clarify the trade-off between predictive accuracy and computational efficiency.

Sensitivity To Early-Exit Confidence Threshold (T)

The dynamic inference mechanism is governed by the entropy-based confidence threshold, τ, which determines whether a sample exists at an intermediate classifier or proceeds to deeper layers. Table V summarizes performance across τ values ranging from 0.1 (aggressive exiting) to 0.9 (conservative exiting).

Table 5 Ablation results for Early-Exit Confidence Threshold (τ) on MobileNetV2

Threshold (τ)	Avg. Latency (ms)	Accuracy (%)	Samples Exiting at Head 1 (%)	Energy Savings (%)
0.1 (Aggressive)	8.42	90.15	82.4	52.1
0.3	10.15	91.80	65.2	44.3
0.5 (Balanced)	12.30	92.45	48.7	35.8
0.7	15.65	92.95	22.1	18.2
0.9 Conservative)	19.80	93.10	8.5	4.5

At low thresholds (τ = 0.1), the model exits 82.4% of samples at the

first head, reducing average latency to 8.42 ms and achieving 52.1% energy savings. This aggressive policy, however, leads to a decrease in accuracy (90.15%). As τ increases, fewer samples exit early, shifting computational load toward the deeper backbone layers. At τ = 0.9, only 8.5% of samples exit at the first head, yielding higher accuracy (93.10%) but limiting latency reduction to 4.5%. A balanced operating point is observed around τ = 0.5, where the model achieves 92.45% accuracy with 35.8% energy savings and 48.7% early exits. This configuration provides a favorable compromise between efficiency and predictive performance. These results demonstrate that the early-exit mechanism functions as a deployment-time control parameter. Without retraining, the model can be tuned to satisfy different power or latency constraints, making it suitable for heterogeneous edge environments (Fan et al., 2024).

Impact of Layer-Wise Pruning Ratios

To assess structural sensitivity, pruning ratios of 30%, 50%, and 70% was evaluated across different regions of the MobileNetV2 backbone. The analysis indicates that early feature extraction layers (Blocks 1–3) exhibit greater sensitivity to parameter removal than deeper bottleneck layers. Applying 50% pruning ratio to early layers resulted in a 3.2% accuracy reduction, whereas the same ratio applied to later expansion layers (Blocks 14–17) produced only a 0.45% degradation. This disparity reflects the importance of preserving low-level spatial features captured in the initial convolutional stages, while later layers contain higher redundancy. These findings validate the selective pruning strategy adopted in this work, where parameter sparsification is applied more aggressively in deeper layers while maintaining higher representational capacity in early layers (Bouzidi, 2024). This structured approach preserves accuracy while achieving meaningful reductions in computational demand.

Comparison With State-Of-The-Art Works

To place the proposed framework within the broader literature of edge-efficient deep learning, this study compares its performance against representative state-of-the-art lightweight and compression-based approaches as summarized in Table 4. Although direct comparisons are constrained by differences in datasets and hardware platforms, the proposed approach demonstrates a highly competitive balance of accuracy and efficiency. For instance, the hybrid model proposed by Musa et al. (Musa et al., 2025) achieved a higher accuracy of 97% by combining pruning, quantization, and knowledge distillation, but resulted in only a 3x reduction in model size. The proposed study offers a more granular trade-off, particularly in energy-per-inference reduction (up to 53% for the 70% pruned model compared to baseline) by integrating dynamic early-exits with structured pruning. Compared with task-specific lightweight models such as the RNN–LSTM architecture proposed by Agarwal & Alam (Agarwal & Alam, 2020), the MobileNetV2-based framework offers greater architectural generality for vision-oriented edge applications while maintaining sub-20 ms inference latency under resource-constrained settings.

Table 6. Comparison of the Proposed Approach with Similar Lightweight Studies

Work	Method/Architecture	Dataset/Task	Accuracy	Efficiency Gain
This Work (Early-Exit)	**EE + Structured Pruning**	**Edge-Bench (Vision)**	**92.79%**	**47.3% Latency Reduction**
Musa et al., (Musa et al., 2025)	Hybrid (Prun+Quant+KD)	Real-life DL Survey	97.0%	3x Parameters Reduction
Agarwal & Alam (Agarwal & Alam, 2020)	Lightweight RNN-LSTM	Human Activity (HAR)	95.78%	Deployed on Raspberry Pi 3
Cui et al., (Cui et al., 2026)	Structured Pruning (RAD)	Edge Deployment	-	7.7x Energy Reduction
Pacheco et al.,(Pacheco et al., 2024)	Early-Exit Inference	Spectrum Class.	-	1.8x Inference Speed-up

Recommendations

Based on the empirical evidence gathered from the head-to-head comparison of static and dynamic optimization paradigms, we provide the following recommendations for researchers and edge-AI engineers:

1. **Prioritize Dynamic Early-Exits for Energy-Autonomous IoT:** For devices powered by batteries or energy-harvesting systems (e.g., smart sensors), the Dynamic Early-Exit strategy should be the default optimization choice. It provides a superior accuracy-per-joule ratio compared to aggressive pruning as validated by the results, which aligns with sustainability goals for Self-Powered Smart Sensors discussed by Cui et al. (2026).
2. **Use Structured Pruning for Deterministic Latency:** In safety-critical applications such as autonomous driving or real-time industrial monitoring, where worst-case execution time must be guaranteed, Static Structured Pruning is recommended. While early-exits provide lower average latency, static pruning offers a deterministic computational graph that simplifies real-time scheduling on heterogeneous edge clusters.

3. **Mandatory INT8 Quantization:** Regardless of the high-level strategy chosen, Post-Training Quantization (PTQ) should always be applied. The 75% reduction in memory footprint observed in this study confirms that quantization is the most cost-effective first step for mobile edge computing deployment.
4. **Sensitivity-Aware Branching:** Engineers should perform layer-wise sensitivity analysis before placing early-exit branches. The study findings suggest that placing exits after resilient layers (e.g., middle bottlenecks in MobileNetV2) avoids accuracy degradation associated with critical layers like the final classifier or initial convolutions.

Hardware-Aware Considerations and Limitations

While the evaluation demonstrates clear performance benefits of both structured pruning and dynamic early-exit strategies on the Raspberry Pi 4, several hardware-aware considerations merit discussion. First, the observed energy savings and latency reductions are specific to the tested configuration; variations in processor load, thermal throttling, and memory availability can influence real-world behavior. Second, dynamic inference introduces variability in execution time, which may affect latency-sensitive applications despite lower average latency. Third, while energy measurements were estimated using average power multiplied by execution time, more precise quantification could be obtained using real-time power monitoring hardware. Finally, the conclusions drawn here may not directly transfer to microcontrollers or ultra-low-power platforms with stricter memory and compute constraints, where additional optimization or model compression strategies could be required.

Conclusion and Future Work

This study presented a systematic comparative analysis of static structured pruning and dynamic early-exit inference for ultra-lightweight deep learning models deployed under edge resource constraints. Unlike prior work that evaluates compression or dynamic execution strategies in isolation, this work examined both paradigms under a unified experimental and hardware-aware evaluation framework. The results demonstrate that static pruning and dynamic early-exit inference occupy distinct and complementary positions in the accuracy–efficiency design space. Empirical findings show that structured pruning achieves reliable and predictable reductions in latency and energy consumption, particularly at moderate pruning levels, while preserving competitive accuracy. Such determinism makes static pruning well-suited for edge applications that require fixed execution behavior and strict worst-case guarantees. On the other hand, early-exit inference enables input-adaptive computation, which delivers superior average-case efficiency and more favorable accuracy–energy trade-off by exploiting uneven difficulty distribution of real-world inputs. These gains persist under post-training quantization, confirming the practicality of dynamic inference for low-power deployment.

Future work could explore hybrid optimization frameworks that combine structured pruning with early-exit mechanisms to jointly leverage deterministic compression and adaptive execution. Extending the evaluation to multiple backbone families and hardware platforms that include microcontrollers and dedicated neural accelerators would provide deeper insights into system-level deployment constraints. Additionally, incorporating runtime-aware exit policies and learning-based threshold adaptation presents a promising direction for improving robustness and efficiency in next-generation edge intelligence systems.

REFERENCES

Adeniyi, O., Sadiq, A. S., Pillai, P., Aljaidi, M., & Kaiwartya, O. (2024). Securing mobile edge computing using a hybrid deep learning method. *Computers*, *13*(1), 25.

Agarwal, P., & Alam, M. (2020). A Lightweight Deep Learning Model for Human Activity Recognition on Edge Devices. *Procedia Computer Science*, *167*, 2364–2373. https://doi.org/10.1016/j.procs.2020.03.289

Alfikri, M. D., & Kaliski, R. (2024). Real-time pedestrian detection on IoT edge devices: A lightweight deep learning approach. *arXiv Preprint arXiv:2409.15740*.

Ateya, A. A., Soliman, N. F., Alkanhel, R., Alhussan, A. A., Muthanna, A., & Koucheryavy, A. (2023). Lightweight deep learning-based model for traffic prediction in fog-enabled dense deployed IoT networks. *Journal of Electrical Engineering & Technology*, *18*(3), 2275–2285.

Biswas, S., & Barma, S. (2023). MicrosMobiNet: A deep lightweight network with a hierarchical feature fusion scheme for microscopy image analysis in mobile-edge computing. *IEEE Internet of Things Journal*, *11*(5), 8288–8298.

Bouzidi, H. (2024). *Efficient deployment of deep neural networks on hardware devices for edge AI* [PhD Thesis]. Université Polytechnique Hauts-de-France.

Chen, Z., Yao, H., Lou, Y., Cao, Y., Liu, Y., Wang, H., & Liu, X. (2021). An empirical study on deployment faults of deep learning based mobile applications. *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, 674–685.

Chersi, F., Bichler, O., Moineau, C., Naud, M., Soutier, L., Templier, V., Allenet, T., Kucher, I., & Lorrain, V. (2025). AIDGE: A Framework for Deep Neural Network Development, Training, and Deployment on the Edge. In *Beyond Horizons–The Rise of the Edge AI Processing Paradigm* (pp. 41–54). River Publishers.

Cordova-Cárdenas, R., Amor, D., & Gutiérrez, Á. (2025). Edge AI in Practice: A Survey and Deployment Framework for Neural Networks on Embedded Systems. *Electronics*, *14*(24), 4877.

Cui, H., Tang, T., & Liu, H. (2026). A Review of Intelligent Self-Powered Sensing Systems Enabling Autonomous AIoT. *AI Sensors*, *2*(1). https://doi.org/10.3390/aisens2010001

Fan, Y., Zhang, L., & Li, P. (2024). A Lightweight Model of Underwater Object Detection Based on YOLOv8n for an Edge Computing Platform. *Journal of Marine Science and Engineering*, *12*(297), 1–19. https://doi.org/10.3390/jmse12050697

Gou, J., Yu, B., Maybank, S. J., & Tao, D. (2021). Knowledge distillation: A survey. *International Journal of Computer Vision*, *129*(6), 1789–1819.

Grzesik, P., & Mrozek, D. (2024). Combining machine learning and edge computing: Opportunities, challenges, platforms, frameworks, and use cases. *Electronics*, *13*(3), 640.

Hassan, M., Davy, S., Zawish, M., Zuber, O. B., & Ashraf, N. (2025). NeuCODEX: Edge-Cloud Co-Inference with Spike-Driven Compression and Dynamic Early-Exit.

arXiv Preprint arXiv:2509.19156.

Hu, Y., Chen, N., Hou, Y., Lin, X., Jing, B., & Liu, P. (2025). Lightweight deep learning for real-time road distress detection on mobile devices. *Nature Communications*, *16*(1), 4212. https://doi.org/10.1038/s41467-025-59516-5

Jang, J., Tulkinbekov, K., & Kim, D.-H. (2023). Task offloading of deep learning services for autonomous driving in mobile edge computing. *Electronics*, *12*(15), 3223.

Jouini, O., Sethom, K., Namoun, A., Aljohani, N., Alanazi, M. H., & Alanazi, M. N. (2024). A survey of machine learning in edge computing: Techniques, frameworks, applications, issues, and research directions. *Technologies*, *12*(6), 81.

Kyem, B. A., Denteh, E. K. O., Asamoah, J. K., Tutu, K. A., & Aboah, A. (2024). Advancing pavement distress detection in developing countries: A novel deep learning approach with locally-collected datasets. *arXiv Preprint arXiv:2408.05649*.

Le Minh, K.-H., Le, K.-H., & Le-Trung, Q. (2020). DLASE: A light-weight framework supporting Deep Learning for Edge Devices. *2020 4th International Conference on Recent Advances in Signal Processing, Telecommunications & Computing (SigTelCom)*, 103–108. https://doi.org/10.1109/SigTelCom49868.2020.9199058

Liu, L., & Xu, Z. (2025). Optimizing lightweight neural networks for efficient mobile edge computing. *Scientific Reports*, *15*(1), 22056. https://doi.org/10.1038/s41598-025-04652-7

Liu, Z., Yuan, D., & Zhu, G. (2025). Automated Blood Cell Detection and Counting Based on an Improved Object Detection Algorithm. *Mathematics*, *13*(18). https://doi.org/10.3390/math13183023

Luo, H., Wei, J., Wang, Y., Chen, J., & Li, W. (2024). An improved lightweight object detection algorithm for YOLOv5. *PeerJ Computer Science*, *8*(23), 2–18.

McClellan, M., Cervelló-Pastor, C., & Sallent, S. (2020). Deep learning at the mobile edge: Opportunities for 5G networks. *Applied Sciences*, *10*(14), 4735.

Merenda, M., Porcaro, C., & Iero, D. (2020). Edge machine learning for AI-enabled IoT devices: A review. *Sensors*, *20*(9), 2533.

Musa, A., Kakudi, H. A., Hassan, M., Hamada, M., Umar, U., & Salisu, M. L. (2025). Lightweight Deep Learning Models For Edge Devices—A Survey. *International Journal of Computer Information Systems and Industrial Management Applications*, *17*(2), 189–206.

Naveen, S., Kounte, M. R., & Ahmed, M. R. (2021). Low-latency deep learning inference model for distributed intelligent IoT edge clusters. *IEEE Access*, *9*, 160607–160621.

Pacheco, R. G., Couto, R. S., & Hoteit, S. (2024). Using Early-Exit Deep Neural Networks to Accelerate Spectrum Classification in O-RAN. *2024 20th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 557–562.

Prebeck, S. S. (2024). *HW-Acceleration for Edge-AI* [PhD Thesis]. Technische Universität München.

Ramya, B., Annam, J. R., Sitamahalakshmi, V., Krishna, V. V. R., Kumar, K. T. R., Jyothi, R. S. S., Suresh, K., & Pallikonda, A. K. (2025). Deploying Lightweight Mobilenet Models On Edge Devices for Energy Efficient Real Time AI in IoT Networks. *Journal of Theoretical and Applied Information Technology*, *103*(11), 4689–4697.

Shankar, V. (2024). Edge AI: a comprehensive survey of technologies, applications, and challenges. *2024 1st International Conference on Advanced Computing and Emerging Technologies (ACET)*, 1–6.

Shuvo, Md. M. H., Islam, S. K., Cheng, J., & Morshed, B. I. (2023). Efficient Acceleration of Deep Learning Inference on Resource-Constrained Edge Devices: A Review. *Proceedings of the IEEE*, *111*(1), 42–91. https://doi.org/10.1109/JPROC.2022.3226481

Theocharides, T., Verhelst, M., Reddy, V. J., & Gousev, E. (2025). TinyML—From Efficient Edge Inference to On-Device Intelligence. *IEEE Design & Test*, *42*(5), 5–7.

Wang, T., Guo, J., Zhang, B., Yang, G., & Li, D. (2025). Deploying AI on Edge: Advancement and Challenges in Edge Intelligence. *Mathematics*, *13*(11). https://doi.org/10.3390/math13111878

Wang, X., Han, Y., Leung, V. C., Niyato, D., Yan, X., & Chen, X. (2020). Convergence of edge computing and deep learning: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, *22*(2), 869–904.

Wulfert, L., Kühnel, J., Krupp, L., Viga, J., Wiede, C., Gembaczka, P., & Grabmaier, A. (2024). AIfES: A next-generation edge AI framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *46*(6), 4519–4533.

Xu, H., Guo, M., Nedjah, N., Zhang, J., & Li, P. (2022). Vehicle and pedestrian detection algorithm based on lightweight YOLOv3-promote and semi-precision acceleration. *IEEE Transactions on Intelligent Transportation Systems*, *23*(10), 19760–19771.

Zaman, S. K. uz, Jehangiri, A. I., Maqsood, T., Haq, N. ul, Umar, A. I., Shuja, J., Ahmad, Z., Dhaou, I. B., & Alsharekh, M. F. (2023). LiMPO: lightweight mobility prediction and offloading framework using machine learning for mobile edge computing. *Cluster Computing*, *26*(1), 99–117. https://doi.org/10.1007/s10586-021-03518-7

Zeeshan, M. (2024). Efficient Deep Learning Models for Edge IOT Devices-A Review. *Authorea Preprints*.

Zhang, Z., Zhao, Y., Chang, M.-C., Lin, C., & Liu, J. (2025). E4: Energy-Efficient DNN Inference for Edge Video Analytics Via Early Exiting and DVFS. *Proceedings of the AAAI Conference on Artificial Intelligence*, *39*(1), 1165–1173.

Zhao, T., Xie, Y., Wang, Y., Cheng, J., Guo, X., Hu, B., & Chen, Y. (2022). A Survey of Deep Learning on Mobile Devices: Applications, Optimizations, Challenges, and Research Opportunities. *Proceedings of the IEEE*, *110*(3), 334–354. https://doi.org/10.1109/JPROC.2022.3153408

Zhu, J., Lou, X., & Ye, W. (2021). Lightweight Deep Learning Model in Mobile-Edge Computing for Radar-Based Human Activity Recognition. *IEEE Internet of Things Journal*, *8*(15), 12350–12359. https://doi.org/10.1109/JIOT.2021.3063504

Zniber, A., Symons, A., Karrakchou, O., Verhelst, M., & Ghogho, M. (2025). Hardware-aware Neural Architecture Search of Early Exiting Networks on Edge Accelerators. *arXiv Preprint arXiv:2512.04705*.