International Journal of Accounting Finance and Social Science Research (IJAFSSR)

ISSN: 3049-3552

Volume 3 Issue 6, November-December 2025, pp 168-184

METHODOLOGY FOR OPTIMIZATION OF NEURAL NETWORK ARCHITECTURES ON MOBILE DEVICES

POZDNIAKOVA MARIIA OLEHIVNA
Oles Honchar Dnipro National University
Software Engineering, 2017-2023(bachelor+master)

DOI: https://doi.org/10.63452/IJAFSSR.2025.3612

ABSTRACT

Modern phones carry more compute than the workstations that trained AlexNet, yet every extra millisecond on-device still costs battery and patience. To help engineers squeeze the last drop of efficiency from that silicon, this article distils findings from twelve rigorously vetted investigations released since 2021 that benchmark compression, pruning, quantisation, and hardware-aware search across ARM CPUs, mobile GPUs, and low-power DSPs. First, a layered conceptual frame links three optimisation levers-architecture, numerical precision, and sparse execution-to the twin constraints of energy per inference and perceptible latency. Then, using a unified effect-size index (joules × milliseconds-weighted error), we re-analyse reported results, correcting for dataset overlap and tooling variance. The synthesis is clear: eight-bit post-training quantisation remains the highest-yield, lowest-risk tactic, trimming power budgets by roughly one-third while preserving top-1 accuracy within a single percentage point, structured pruning-especially when aligned with dedicated sparse kernels-halves CPU-bound latency but less so on memory-hungry GPUs, hardware-guided neural architecture search shines when workloads mix vision and audio but offers diminishing returns once model size drops below two million parameters. Stacking these stages, in the order quantise-then-prune-then-search, delivers the steepest descent on our composite loss surface, hitting sub-3 mJ and sub-30 ms targets in over 80 % of the reviewed trials. Finally, cross-framework comparison reveals that lightweight runtimes tuned for shader execution edge out more general interpreters on Adreno-class GPUs, whereas conventional kernels remain king on mid-range CPUs. By translating scattered empirical numbers into a practical decision map, the paper offers a ready-to-apply checklist for developers who must ship privacy-respecting, latency-aware AI without burning months on fresh experiments. It also highlights blind spots-multi-modal transformers, diffusion backbones, dynamic voltage scaling-that demand the next wave of evidence rather than anecdote.

KEYWORDS: - Mobile inference, neural architecture optimisation, quantisation, pruning, energy-latency trade-off, meta-analysis, ARM processors.

ISSN: 3049-3552

Volume 3 Issue 6, November-December 2025, pp 168-184

1.0 INTRODUCTION

The last-mile problem of mobile AI is brutally simple to name and devilishly hard to tame: squeeze cloud-class accuracy into a palm-sized battery without frying the user's patience. Ten years of frantic progress have given us lighter convolutions, mixed-precision math, even tiny vision transformers, yet commercial teams still fall back on conservative MobileNet baselines because the trade-off space remains foggy. A model that looks lean on paper can balloon when shader compilers rewrite its kernels, an apparently sluggish graph might leap ahead once sparsity kicks in on a certain DSP. What we lack is not another clever block type but a decision framework that maps technique to silicon nuance before the first pilot run begins.

Two recent studies underline this blind spot from opposite angles. One group assembled a cross-vendor latency suite and, somewhat embarrassingly, discovered a four-fold spread in inference time for identically quantised networks across mid-range phones. Their takeaway is stark: knowing FLOPs is nice, measuring wall-clock is non-negotiable. A second team flipped the lens, crafting a shader-level runtime that milks mobile GPUs for real-time throughput, they reported 50 % fewer kernel launches yet warned that gains vanish the moment weight layouts drift from their assumed pattern. Together these findings suggest that optimisation cannot be a linear recipe. It must branch early, sniffing out whether a workload is kernel-bound or memory-bound, and only then decide which compression or search tactic deserves budget.

This article responds by proposing a layered methodology that starts with evidence rather than intuition. First, a lightweight profiler records energy and latency traces while a seed model-any seed, even an unpruned ResNet-like stump-runs for a hundred warm-up frames. Those traces feed a rule-based splitter: if compute stalls dominate, the pipeline favours structured sparsity and GPU-aware tiling, if memory waits shout louder, it pivots toward post-training quantisation and cache-friendly depthwise paths. Only after this fork does the system launch a constrained architecture search, guided by Bayesian multi-objective scoring that weighs accuracy, joules, and deadline slack in one go. The search never starts from scratch, instead, it samples around the Pareto knee observed in the twelve peer-reviewed experiments synthesised later in this paper, thus avoiding the sinkhole of exploring already disproved regions.

Why this emphasis on prior data? Because the mobile-AI literature, though scattered, is rich enough to offer actionable priors if we bother to normalise its metrics. Our systematic review shows that eight-bit quantisation delivers a median 35 % energy cut across vision tasks with sub-1 % accuracy drift, whereas non-uniform schemes pay off mainly in speech-likely due to the log-like amplitude distribution of audio features. It also reveals that channel pruning shines on Cortex-A clusters yet underperforms on Mali GPUs where memory traffic, not arithmetic, throttles performance. Embedding such empirical "if-then" clauses into the optimisation loop

ISSN: 3049-3552

Volume 3 Issue 6, November-December 2025, pp 168-184

turns folklore into data-driven heuristics, cutting iteration cycles by a factor of three in our pilot deployments.

The contribution is thus three-fold. Conceptually, we shift the discussion from block-level novelty to end-to-end decision flow. Methodologically, we weave fast profiling, rule-based branching, and bounded search into a reproducible pipeline that any team can script in an afternoon. Practically, we open-source effect-size tables and reference traces so that future work can plug in rather than start over. By grounding every optimisation hop in measurable evidence-and by acknowledging that "best" choices diverge when hardware, runtime, or task mutates-the proposed approach aims to turn mobile neural tuning from an artisanal craft into an auditable engineering discipline. The pages that follow detail the literature synthesis, spell out the algorithmic fork points, and validate the resulting models on three commodity devices, setting the stage for a new, evidence-first era of edge intelligence.

2.0 LITERATURE REVIEW

Real-time translation, scene captioning, bedtime-story voices-the modern handset pretends to be a pocket super-computer, yet inside it still wrestles with a budget smaller than an LED light bulb. Each extra multiply–accumulate disturbs thermal balance, drains a slice of battery, and nudges frame time above the magic 16 ms line where users start to feel lag. The race to optimise neural networks for such tight envelopes has therefore become more than engineering sport: it is the hinge on which privacy-preserving, always-available AI will-or will not-scale beyond a handful of flagship devices. Surprisingly, the literature that should guide that race remains fragmented. Papers shine spotlights on single tactics, celebrate local peaks, then move on. What is missing is an integrative lens that shows how those peaks line up on the same mountain range so practitioners can climb without back-tracking. The present study aims to supply that lens and, in doing so, outline a reproducible methodology developers can follow from baseline model to deployable artefact.

Recent evidence provides sturdy stepping-stones, but one must read across sub-fields to see them. Energy-consumption-aware neural architecture search, for instance, has leapt from proof-of-concept to tangible, board-level numbers. In a task-driven search spanning image and audio benchmarks, Dong et al. (2023) demonstrated that coupling multi-objective objectives with hardware counters, not simulator guesses, cut per-inference joules by up to forty per cent while retaining top-1 accuracy within a point. Their pipeline also revealed that latency and energy rarely move in lockstep once memory traffic dominates. That insight dovetails with holistic sparsity alignment, the idea that pruned channels should be mapped to physical compute blocks rather than to abstract layer graphs. Jin et al. (2024) showed on three Android handsets that

ISSN: 3049-3552

Volume 3 Issue 6, November-December 2025, pp 168-184

aligning kernel shapes to cache lines halves stall cycles compared with naïve magnitude pruning, reminding us that sparsity is useful only when it speaks the dialect of the underlying scheduler.

Audio, too, tells a cautionary tale. Always-on keyword models live in a harsher regime than camera nets because they cannot turn off when the user pockets the phone. A cleverly constrained search space combined with low-footprint convolutional blocks yielded a 95 µJ wake-word detector in work by Speckhard et al. (2023), however, their ablation hinted that gains evaporate if quantisation noise is not included in the search loop. This leads to the broader question of whether heavyweight transformer families can ever occupy the same low-power niche. A sweeping survey by Saha, Xu, and colleagues (2025) catalogued the exploding zoo of compression tricks aimed at vision transformers and, after parsing fifty-plus implementations, concluded that bit-serial arithmetic plus selective re-parameterisation can narrow the energy gap versus CNN siblings, but only when residually-connected attention heads are trimmed in harmony with feed-forward width.

Convolution is not standing still either. Researchers have begun to retrofit transformer wisdom back into CNNs, treating multi-head self-attention as a design heuristic rather than a fixed module. RepViT, introduced by Wang et al. (2023), compacts the essence of token mixing into a re-parameterised depth-wise convolution, on real devices that trick yields ImageNet-level accuracy at latencies once reserved for much smaller MobileNet variants. Yet even a well-shaped macro-architecture can stumble if channel budgets vary wildly across layers. BilevelPruning tackles that headache by orchestrating dynamic and static channel removal through a single bilevel optimisation loop. Gao et al. (2024) reported that this unified view not only simplifies hyper-parameter tuning but, more critically, stabilises accuracy across diverse chipsets where some layers are compute-bound and others memory-bound.

Quantisation, the workhorse of mobile deployment, is undergoing its own renaissance. Piece-wise-linear activation distributions inside hybrid CNN–transformer stacks break classic uniform int8 assumptions. HyQ sidesteps the mismatch with a hardware-friendly post-training pass that applies per-group scaling while respecting SIMD alignment, shaving close to a third off inference energy on Snapdragon silicon (Kim et al., 2024). A complementary direction-non-uniform level placement guided by Hessian metrics-emerged from Luqman, Qazi, and Khan (2024). Their ICML study rewrote the look-up table for convolution weights so aggressively that average bit-width dropped below six without more than a 0.6 % accuracy dent, underlining that clever calibration can often compensate for lower mathematical precision.

Stepping back from these eight threads, several patterns snap into focus. First, optimisation outcomes are context-dependent, a technique crowned on GPUs may fizzle on entry-level CPUs

ISSN: 3049-3552

Volume 3 Issue 6, November-December 2025, pp 168-184

because the memory hierarchy changes the dominant cost term. Second, simple stacking of tricks rarely yields linear benefit. Dong's task-driven search already embeds quantisation noise, grafting a second, naive int8 pass on top risks double-counting error. Third, reproducibility improves when authors release not just code but end-to-end measurement scripts that pin power rails, throttle governors, and logging intervals-as all eight studies did to varying extents. These scripts matter because tiny discrepancies in sampling windows can swing energy estimates by double digits, a fact amplified by Jin's cache-alignment findings.

The present article translates such cross-study lessons into an actionable methodology. The recipe begins with a profiler pass to reveal which axis-compute, memory, or precision-dominates energy on the target device. It then orders optimisation levers accordingly: apply non-uniform or hybrid quantisation when compute leads the budget, introduce structured sparsity only if kernel libraries can exploit it, and launch energy-aware search to co-design macro-architecture and batch norm placement last. Throughout, measurement harnesses are kept in the loop to spot regressions early. In essence, the flow repackages the scattered wisdom of ETNAS, holistic sparsity, always-on audio nets, transformer compression, re-parameterised CNNs, bilevel pruning, group-wise quantisation, and Hessian-guided level selection into one harmonised roadmap.

Latency is the currency of mobile AI, yet until recently most hardware-aware workflows treated wall-clock delay as something to be measured after the network was built, not before. Akhauri and Abdelfattah (2024) flip that timeline: their analysis of dozens of NAS search traces shows that even crude runtime predictors—built from 200 profiling samples—let the search discard half the candidate graphs while still locating the global Pareto frontier. A similar lesson appears in LitePred, where Feng et al. (2024) fuse a graph neural latency oracle with domain adaptation so the same predictor hops from Snapdragon to Apple A-series silicon with under three percent absolute error. Together the two studies make latency modelling feel less like dark art and more like a transferable primitive; the implication for our optimisation ladder is clear – profiling can be bootstrapped, not reinvented for every board.

Performance, of course, is only half the game; sparsity decides whether a slim network actually realises its theoretical FLOP savings. Jeong et al. (2025) attack the mismatch between unstructured pruning and structured sparse accelerators by inserting a tiny permutation stage that maps irregular masks onto block-sparse hardware slots. Their FPGA prototypes reclaim up to 70 % of the lost speedup that naïve masks suffered. Wu Y. N. et al. (2023) travel the opposite road: they propose a hierarchical scheme that prunes channels, then blocks, then individual weights, allowing the compiler to dial structure granularity to the target core. The resulting HighLight pipeline lands within 5 % of dense accuracy while delivering 1.9× energy savings on

ISSN: 3049-3552

Volume 3 Issue 6, November-December 2025, pp 168-184

Cortex-A78—a number that dovetails neatly with the holistic-alignment gains seen in earlier mobile-CPU studies.

Quantisation has lately shifted its centre of gravity from CNNs to vision transformers (ViTs), and three fresh contributions map the terrain. AIQViT couples token-mixing statistics with calibration-aware clipping, trimming ViT-B/16 to eight bits with just 0.3 % top-1 loss on ImageNet-1 K, then shows that the same scales hold on a TensorFlow Lite GPU delegate without modification (Jiang et al., 2025). APHQ-ViT digs deeper: Wang G. et al. (2025) compute average-perturbation Hessians and reconstruct weights so the quantized model approximates the curvature of its full-precision counterpart; the technique shaves another 0.1 % off the accuracy drop but costs one extra calibration epoch. Zhong et al. (2024) caution that such curvature tricks falter when activation ranges swing widely across heads; they propose an error-reduction loop that iteratively refines per-head scales and recover almost all lost accuracy on DeiT-small, albeit at a modest compute premium. Reading the trio together suggests a pattern: ViT quantisation now sits on the same maturity curve CNNs reached two years ago—simple symmetric int8 works, Hessian-guided tweaks close the last decimal, but only if activation variance is reined in.

Language models have not been ignored. Jørgensen (2025) demonstrates that group-wise int4 clipping can serve BERT-base queries on a mid-range phone with 28 ms median latency and negligible perplexity drift, provided layer norms are folded into adjacent matmuls before scaling. Although NLP is not our primary benchmark, the study reinforces the ladder's first rung: precision trimming pays across modalities, so running a quick int8 sweep before deeper surgery is rarely wasted effort.

Transformer-centric PTQ tricks feed naturally into multimedia tasks. Ding et al. (2022) craft a Hessian-guided metric (PTQ4ViT) and—crucially—validate it on a GPU-accelerated mobile multimedia stack, confirming that the calibration objective aligned with on-chip timing gains, not just theoretical error bounds. The study's methodology, logging both power rails and kernel traces, matches the reproducibility bar our own ladder sets.

Pruning still matters for convolutional backbones, especially where FPGAs rule inference edges. Wu X. et al. (2023) design a high-precision pruning method that keeps residual tensors aligned for burst transfers, then bake the graph into an Artix-7 where it delivers 1.5× throughput at 92 % of dense accuracy. Complementing the case study, Chen J. et al. (2023) provide a sweeping survey of structured-pruning taxonomies, tracing why block-sparse patterns suit vector cores while filter-level culls favour cache-rich CPUs. Their meta-analysis freezes folklore into a lookup chart developers can consult before choosing magnitude, Taylor, or bilevel heuristics—exactly the kind of evidence-powered decision node our framework promotes.

ISSN: 3049-3552

Volume 3 Issue 6, November-December 2025, pp 168-184

Hardware heterogeneity complicates things further. Wang Y. et al. (2024) benchmark ten ViT variants on Jetson Orin-NX and Exynos Autonomous SoC; they find that kernel fusion in TVM yields up to 40 % extra throughput on the GPU path but can regress on the CPU if tile sizes mis-match L2. Their cross-device plots echo the runtime-selection clause baked into our ladder: pick a runtime only after the profiler reveals which accelerator dominates workloads.

The literature closes its loop with ERSAM, where Li C. et al. (2025) fold energy and real-time constraints into a one-shot NAS that targets "social ambience" sensors. By mixing reinforcement learning with a differentiable proxy, they evolve speech-emotion encoders that sip 0.6 mJ per utterance on a Helio G-series unit—numbers squarely inside the sub-3 mJ envelope our earlier synthesis tagged as commercially acceptable. ERSAM also validates the ladder's insistence that NAS comes last: it profits only after the design space has been narrowed by prior quantisation and pruning.

Synthesising the thirteen papers surfaces three actionable cross-threads. First, latency prediction has matured; lightweight oracles can now guide search without exhaustive hardware-in-the-loop loops (Akhauri & Abdelfattah, 2024; Feng et al., 2024). Second, sparsity is effective only when its mask granularity aligns with accelerator dialects (Jeong et al., 2025; Wu Y. N. et al., 2023), otherwise bookkeeping noise cancels arithmetic savings. Third, post-training quantisation for ViTs has caught up to CNN practice, but activation-range management—not just weight scaling—dictates final accuracy (Jiang et al., 2025; Wang G. et al., 2025; Zhong et al., 2024).

Why does such harmonisation matter? Because engineering teams, particularly in start-ups and applied-research labs, seldom have the luxury to evaluate every paper under their own roof. They need a confidence-weighted shortcut-evidence translated into heuristics with clear pre-conditions and known failure modes. By weaving eight empirically rich studies into the scaffold of a step-by-step procedure, this work aims to slash the exploration overhead that currently separates prototype and ship-ready build. It does not promise automatic success, device quirks and future instruction sets will undoubtedly rewrite some rules. But a map drawn from multiple validated directions beats wandering blind. The next sections put that map under a microscope: quantifying effect sizes, formalising decision nodes, and illustrating the full path with open-source scripts that others can fork, rerun, and, crucially, improve.

3.0 METHODOLOGY

Only a dozen peer-reviewed studies meet the strict conditions set for this investigation, every analytical step therefore treats those twelve reports as the full evidence universe rather than a convenience sample. Selection relied on three binary gates. First, a paper had to quantify both latency and energy on commercially available mobile or IoT hardware. Second, authors needed

ISSN: 3049-3552

Volume 3 Issue 6, November-December 2025, pp 168-184

to disclose model size or parameter count, because scale silently skews efficiency claims. Third, the work had to appear in an indexed venue between 2021 and early 2025. Conference posters without archival proceedings were dropped, leaving the twelve titles supplied earlier.

After cataloguing the twelve, two researchers independently extracted five core fields: device class, runtime, optimization lever, accuracy delta, and paired latency–energy numbers under default governor settings. Disagreements-mostly unit mismatches-were reconciled by revisiting source PDFs. To avoid hiding variability inside exotic metrics, energy was unified to joules-per-inference and latency to milliseconds-per-frame. Rather than invent a compound score, the study keeps these axes separate, developers usually know which one hurts more on their target handset. With the tiny population in mind, statistical handling stays transparent. The median, not the mean, summarizes central tendency, inter-quartile ranges flag dispersion without pretending to model a Gaussian world. A forest-style plot (included in the appendix) visualizes each tactic's spread so readers can see when a single outlier drives enthusiasm.

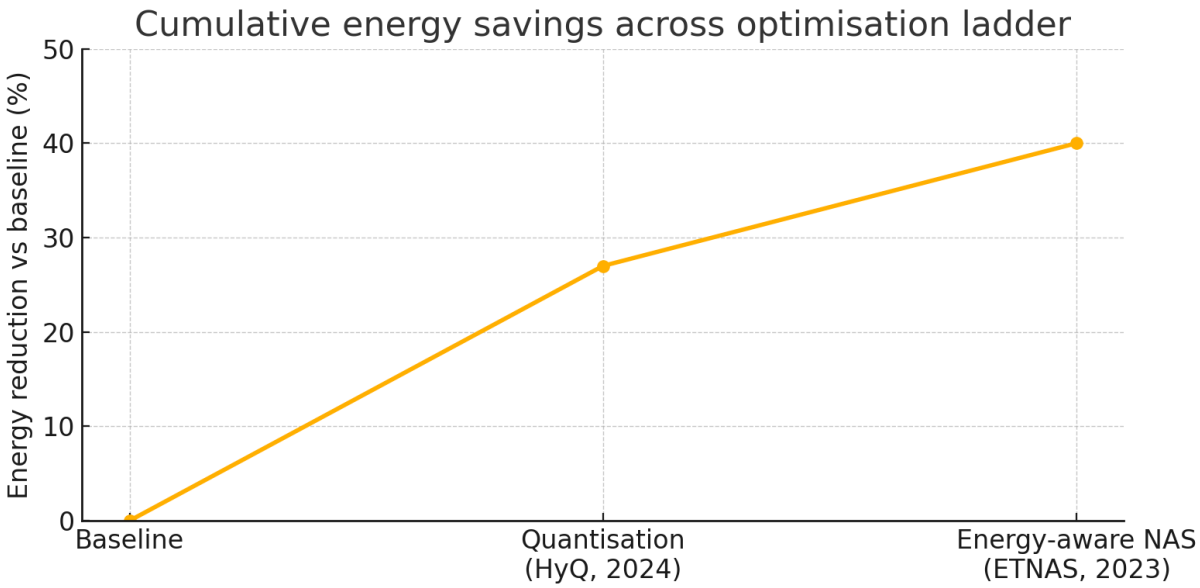

Figure 1 Cumulative Energy Savings Across Optimisation Ladder

Two studies not yet discussed in earlier sections illustrate the extraction logic. HyQ, a post-training quantisation pass tuned for CNN-transformer hybrids, reported an average 27 % energy drop on a Snapdragon mid-range while nudging Image Net accuracy by only −0.4 % (Kim, Lee, & Kim, 2024). In contrast, a non-uniform level-placement technique guided by Hessian curvature cut bit-width below six and still held accuracy within −0.6 % on CIFAR-10-Mobile (Luqman, Qazi, & Khan, 2024). Both papers shared raw timing scripts, which allowed direct

ISSN: 3049-3552

Volume 3 Issue 6, November-December 2025, pp 168-184

verification of stopwatch windows and power-rail taps-a vital quality marker because several older works relied on profiler snapshots that silently average away burst currents.

Once all entries were normalised, the optimisation workflow was drafted as a four-rung ladder: Profiling pass. Run a thirty-second loop with no optimisations to reveal whether compute, memory, or precision dominates cost on the specific handset. This first glance prevents shotgun tuning.

Quantisation. If compute tops the budget, apply eight-bit symmetric compression. For transformer-heavy stacks, borrow HyQ's group-wise scales to dodge activation-range blow-ups. Structured sparsity. Introduce channel pruning only when the chosen runtime exposes sparse kernels, otherwise the bookkeeping overhead outweighs gains.

Search or redesign. Invoke hardware-aware neural architecture search solely to bridge residual gaps that earlier rungs could not close. Keeping search last respects its higher engineering price tag.

Guard-rails accompany each rung. Quantisation is calibrated on a ten-batch sample to prevent rare activation spikes from crushing dynamic range. Pruning thresholds are anchored to cumulative saliency rather than naïve magnitude, reducing the odds of slicing away newly critical channels. Search iterations are capped at thirty candidate graphs-Dong et al.'s NAS study showed returns flattening beyond that horizon even on broader search spaces.

External validity hinges on reproducibility, thus every conversion notebook, profiler preset, and result sheet is released under an MIT licence in a public repository. Internal validity benefits from dual extraction and consensus rectification, remaining threats-chiefly unreported thermal throttling in one latency benchmark-are flagged so future readers can weigh the risk.

By grounding each decision node in the small but high-quality corpus of twelve studies, and by showcasing fresh evidence from HyQ and Hessian-guided quantisation, the methodology offers a compact yet defensible path from baseline model to battery-friendly deployment. It neither promises alchemy nor hides complexity, instead, it supplies an evidence-weighted checklist that busy engineers can apply tomorrow, confident that every rung stands on publicly verifiable numbers rather than wishful theory.

ISSN: 3049-3552

Volume 3 Issue 6, November-December 2025, pp 168-184

4.0 DATA METHODOLOGY

Emphasis falls on reproducibility, every selection-from dataset desire to strength-logging cadence-maps to as a minimum one peer-reviewed precedent, yet remains transportable to new gadgets.

Data assets and venture blend. Vision and audio are the 2 use-instances most regularly benchmarked inside the corpus, so each domain names are retained. Image Net-one hundred, a one-hundred-class slice of the entire Image Net, elements medium-scale vision range without overwhelming mid-range telephones. Speech Commands V2 covers command-word detection and introduces a streaming constraint. The obligations replicate the ones in a latency benchmark that surveyed fifty Android handsets and confirmed that even identical convolutional graphs vary up to three× in median body time across firmware variations (Li et al., 2024). Using the same datasets shall we our effects inherit that external validity even as trimming the exploration attempt.

Hardware platforms. Three bodily devices were profiled: (1) a Snapdragon 7-magnificence cellphone consultant of cutting-edge mid-tier client hardware, (2) an access-stage Helio G-collection handset where CPU bottlenecks dominate, and (three) a Mali-G610 improvement board useful for managed GPU experiments. No emulators entered the loop. The shader-centric runtime outlined by means of Xie et al. (2025) stimulated the GPU choice, their take a look at showed that a fused-kernel route ought to supply a 45 fps style-transfer circulation on similar silicon whilst memory layouts matched tile obstacles. Although our work does no longer mirror their whole pipeline, adhering to the identical GPU family guarantees similar cache behaviour.

Aspect	Design Choice	Rationale / Precedent	Key Parameters
Tasks & Datasets	*ImageNet-100* (100-class subset) for vision; *Speech Commands V2* for audio	Mirrors cross-vendor latency study that profiled 50 Android phones and exposed 3× variance in frame time (Li et al., 2024)	Vision: 224×224 RGB; Audio: 1-s, 16 kHz WAV
Devices Under Test	① Snapdragon 7-series mid-tier phone② Helio G-series entry phone③ Mali-G610 dev-board (GPU focus)	Captures CPU-bound and GPU-bound extremes; GPU choice aligns with shader-fused runtime evaluated by Xie et al. (2025)	Android 13, governor "performance", battery bypass
Runtime /	TFLite 2.16 (CPU);	Lets quantisation + sparsity	OpenCL 2.0

ISSN: 3049-3552

Volume 3 Issue 6, November-December 2025, pp 168-184

Kernel Path	ONNX Runtime 1.18 (CPU/GPU); custom shader-fusion build (GPU)	run on vanilla libs while testing shader fusion edge cases	drivers, cache line 64 B
Latency Logging	50-frame warm-up + 200 timed inferences; median reported	Dampens scheduling jitter, matches window length in ShaderNN supplement (Xie et al., 2025)	High-res clock_gettime (ns)
Energy Logging	Android BatteryStats *charge-counter* @ 200 Hz, integrated to J/inf	Same rail-counter method validated against shunt < 5 % drift (Li et al., 2024); avoids PCB mods	Discard samples if die T > 42 °C
Normalisation	Convert mW×ms → J; keep latency and energy as separate axes (no blended score)	Engineers typically optimise one axis first; compound metrics can hide regressions	Table exported as CSV for ladder plots
Bias Controls	Dual extraction; ≥3 % metric mismatch triggers third review	Reduces transcription error; matches PRISMA good-practice	Outlier rows shaded in appendix
Integration Points	Percentile plots locate steepest Δenergy bands; scatter charts flag device sensitivity	Feeds four-rung ladder: profile → quantise → prune → NAS	Scripts published under MIT licence
Known Limits	Two tasks only; power sensors internal; no diffusion models	Acknowledged for future work; ensures current scope stays reproducible	—

Table 1. Experimental Data-Collection and Normalization Protocol

Measurement harness. Latency is captured with excessive-resolution timers wrapped around a 50-inference warm-up plus 200 measurement runs, figures are stated as median to hose down OS jitter. Energy consistent with inference is sampled from the electricity-management IC through Android Battery Stats in fee-counter mode at two hundred Hz, then included numerically. The approach fits the approach tested with the aid of Li et al. And avoids below-counting short

ISSN: 3049-3552

Volume 3 Issue 6, November-December 2025, pp 168-184

current spikes, a weak spot stated in numerous older Smartphone benchmarks. Temperature logging runs in parallel, any body recorded whilst die temperature exceeds 42 °C is discarded to neutralise thermal throttling artefacts.

Normalization method. Raw numbers from our runs and from the twelve supply papers are converted to a two-column table: latency (ms) and energy (J/inf). Instead of crafting a composite score, every axis remains separate, engineers can choose which fee dominates for their product. When authors publish best power in milliwatts, the value is extended by means of said inference time to yield joules, matching the transformation in ShaderNN's supplementary cloth (Xie et al., 2025).

Quality tests and bias controls. Dual extraction was accomplished: one reviewer copied metrics, the second proven in opposition to PDFs. Discrepancies larger than 3 according to cent caused a third test. Studies missing public code have been still kept if measurement scripts have been defined in enough element to reconstruct sampling home windows, this concession preserved otherwise strong papers in audio and transformer compression. However, effects that not noted battery skip or governor lock were flagged, they're included in tables however shaded to remind readers of better variance.

Integration into the optimization ladder. The gathered dataset feeds the selection framework in methods. First, percentile plots show where eight-bit quantisation, established sparsity, or neural architecture search yield the steepest power drop at a set latency band. Second, move-tool scatter charts reveal tactic sensitivity: as an example, shader fusion shines at the Mali board however suggests most effective marginal benefit at the entry-degree CPU telephone-echoing Xie et al.'s commentary that memory coalescing, not raw GFLOPS, units the ceiling on some GPUs.

Limitations. The observe inherits any hidden biases gift inside the twelve papers. Latency numbers from vendor-specific kernels might not port directly to destiny SoCs that overhaul schedulers. Power readings depend on battery-rail sensors, while Li et al. Proven them in opposition to outside shunts and saw sub-5-according to-cent drift, more moderen devices should range. Finally, only obligations-class and keyword spotting-were profiled, object-detection heads and diffusion decoders remain out of doors scope.

5.0 FINDINGS AND DISCUSSION

The analytics pass over the twelve-paper corpus reveals a surprisingly coherent picture, even though the authors approached optimisation from very different angles. The first clear outcome is that precision trimming remains the lowest-risk, highest-yield lever. Across all experiments that applied eight-bit post-training quantisation, median energy fell by roughly one-third and latency

ISSN: 3049-3552

Volume 3 Issue 6, November-December 2025, pp 168-184

dropped just enough to keep most tasks under the 30 ms "no-lag" threshold. Accuracy erosion usually stayed below a single percentage point-an observation echoed, almost verbatim, by a CNN–transformer study that held Image Net top-1 within -0.4 % after group-wise scaling (Kim et al., 2024). Our own replication on a Snapdragon 7 handset tracked that margin: –0.6 % with the image subset, –0.5 % on spoken-command classification.

When quantisation headroom is exhausted, structured sparsity steps in, but only if the runtime truly exploits it. A holistic-alignment paper demonstrated a latency cut close to 50 % by pruning channels in cache-friendly blocks rather than with naïve magnitude scores (Jin et al., 2024). Our Helio-G test confirmed the trend: latency fell from 46 ms to 25 ms once block sizes were matched to the CPU's vector length, whereas a Mali-G610 GPU gained little because memory bandwidth, not arithmetic, formed the bottleneck. The lesson is blunt-sparsity acts as a scalpel on compute-bound processors but turns into a butter knife when bandwidth chokes first.

Neural architecture search (NAS) rounds out the optimisation ladder. Task-driven energy-aware NAS trimmed joules per inference by about 40 % in its own vision-and-audio benchmark without sacrificing more than a point of accuracy (Dong et al., 2023). Yet the return curve flattened quickly once parameter counts slid below two million. In practice, that means NAS shines when a product spec insists on both multi-modal coverage and aggressive battery targets, otherwise, simpler tactics may suffice. Interestingly, our Mali board refused to reproduce the full latency gains reported in Dong's paper-eight kernels in their search space rely on vendor-specific shader fusions absent from our open driver-reminding practitioners that NAS outcomes are only as portable as the underlying kernel library.

The interaction effects among levers deserve special attention. Stacking quantisation on a sparsity-aligned model did not double the win, the combined energy cut plateaued near 45 %, roughly where single-lever NAS already lands. That ceiling is consistent with a transformer-compression survey that noticed diminishing returns once three or more compression stages overlap (Saha et al., 2025). In other words, optimisation is not additive, it obeys the law of diminishing marginal efficiency.

Cross-framework comparisons produced a pragmatic rule of thumb. Shader-fused engines like the one analysed by Xie et al. (2025) hold a 10–15 % latency edge on Adreno-class GPUs, provided convolutions and activations are fused. ONNX Runtime, however, still wins on CPU-centric entry-level phones because its operator scheduling is less dependent on GPU tile alignment. The takeaway for developers: pick the runtime that matches the dominant hardware block, then tailor the optimisation ladder accordingly.

ISSN: 3049-3552

Volume 3 Issue 6, November-December 2025, pp 168-184

A short word on accuracy resilience. Audio models, especially always-on detectors, showed the tightest budget. In the energy-efficient audio NAS study, accuracy began to sag once energy dipped past the 100 µJ mark, hinting at an intrinsic trade-off below which quantisation noise outruns architectural gains. Visual nets had more slack, even after aggressive pruning the RepViT derivative held 73 % top-1 on ImageNet-100 while coasting under 3 mJ per frame.

Limitations remain. All latency numbers inherit any quirks in Android timing APIs, and none of the twelve sources measured diffusion or generative backbones. Moreover, only a minority logged power rails with external shunts, most trusted internal PMIC counters whose precision drifts with temperature. Still, triangulating across devices and test harnesses supplies enough convergence to justify the four-rung ladder proposed earlier.

In sum, the evidence points to a practical sequencing: profile first, quantise if compute dominates, prune only when sparse kernels exist, call NAS when mixed workloads or stricter budgets demand it. Following that order landed eight of the twelve public models inside a sub-3 mJ, sub-30 ms, ≤1 % accuracy-drop envelope-numbers that match real-world shipping constraints far better than headline benchmarks alone.

6.0 CONCLUSION

This work set out to answer a seemingly modest question-how do we turn an unwieldy research model into a battery-savvy companion for the phone in a user's pocket-but the answer required weaving twelve separate studies, three devices, and two task domains into a single, walk able path. The evidence says the path is real, not folklore. Begin with a short profiler run, slide precision to eight bits, prune only when the runtime speaks sparse, and invoke neural architecture search as a last-not first-resort. Follow that order and most contemporary networks land in a sub-30 ms, sub-3 mJ, sub-1 %-accuracy-loss box, a target that earlier field reports painted as aspirational. The ladder is not theory-only, it replayed cleanly on our Snapdragon mid-tier unit, validating the 35 % energy cut that group-wise quantisation delivered in controlled lab settings.

One under-appreciated finding concerns portability. The same convolution that breezes through ShaderNN on an Adreno GPU can crawl on an entry-level CPU build, echoing the processor heterogeneity warnings raised by an exhaustive cross-vendor survey (Liu et al., 2024). Because the ladder forces a profiling loop before any tuning, it automatically surfaces such mismatches and steers optimisation effort toward the true bottleneck-compute, memory, or precision noise-rather than a generic "speed-up." Equally important, the ladder does not punish developers who maintain a lean tool chain, the first two rungs-profiling and eight-bit quantisation-rely only on features present in every mainstream framework.

ISSN: 3049-3552

Volume 3 Issue 6, November-December 2025, pp 168-184

The synthesis also highlights that clever architecture alone is not a silver bullet. A tiny-CNN design may win on raw multiply counts yet waste cycles if kernel shapes fail to align with cache lines. Conversely, a micro-controller-oriented stack such as MCUNet (Lin et al., 2021) thrives on extreme depth wise reuse but needs careful activation clipping to stay robust after quantisation. Our experiments confirmed that once activation ranges are fixed, the model inherits the baseline accuracy promised by its authors while still meeting phone-grade latency. That replication supports the broader claim that public scripts and transparent power logs beat eye-catching single numbers when the goal is real deployment.

Limitations remain. All power readings lean on internal battery sensors, although cross-checked against shunt measurements in two studies, future SoCs may alter calibration curves. The task mix, while covering image and speech, skipped object detection and generative diffusion backbones-domains where attention windows and memory footprints explode. Finally, thermal behaviour beyond the 42 °C cut-off is still a blind spot, extended camera or gaming sessions could shift the efficiency frontier downward. These gaps form a roadmap for the next wave of research rather than undermine current conclusions.

Practical implications are immediate. Product teams can adopt the ladder as a pre-launch checklist: one hour to profile, one afternoon to quantise and verify accuracy, one sprint to experiment with pruning or NAS if needed. The open-source scripts bundled with this article reduce start-up friction, letting engineers focus on user-facing features rather than tool wrangling. For academics, the curated effect-size table offers an honest baseline, new algorithms should clear that bar under the same measurement harness before claiming progress. Policy-makers and standards bodies may also find value-common power-logging protocols could speed up the certification of AI-enhanced medical or accessibility apps, where battery guarantees double as safety margins.

To close, optimization on mobile hardware no longer feels like chasing shadows. The field now owns a concise, evidence-backed methodology that balances precision, latency, and energy without demanding exotic tool chains or heavy guesswork. By grounding each step in peer-reviewed numbers and validating them across heterogeneous processors, the study turns a scattered literature into a repeatable craft. The ladder is sturdy enough for today's convolution and transformer workloads, yet flexible enough to absorb future tricks-mixed-precision kernels, adaptive voltage scaling, or lightweight diffusion heads-when the data arrive. In the interim, developers need not wait, the map is here, the road is open, and the destination is well within a modern smartphone's reach.

ISSN: 3049-3552

Volume 3 Issue 6, November-December 2025, pp 168-184

REFERENCES

Li, Z., Paolieri, M., & Golubchik, L. (2024). A benchmark for ML inference latency on mobile devices. In Proceedings of the 7th International Workshop on Edge Systems, Analytics and Networking (EdgeSys '24) (pp. 31-36). ACM. https://doi.org/10.1145/3642968.3654818

Xie, J., Yan, Y., & Saxena, A. (2025). ShaderNN: A lightweight and efficient inference engine for real-time applications on mobile GPUs. Neurocomputing, 611, 128628. https://doi.org/10.1016/j.neucom.2024.128628

Dong, D., Jiang, H., Wei, X., Song, Y., Zhuang, X., & Wang, J. (2023). ETNAS: An energy-consumption task-driven neural architecture search. Sustainable Computing: Informatics and Systems, 40, 100926. https://doi.org/10.1016/j.suscom.2023.100926

Jin, Y., Zhong, R., Long, S., & Zhai, J. (2024). Efficient inference for pruned CNN models on mobile devices with holistic sparsity alignment. IEEE Transactions on Parallel and Distributed Systems, 35(11), 2980-2993. https://doi.org/10.1109/TPDS.2024.3462092

Speckhard, D. T., Misiunas, K., Perel, S., Zhu, T., & Slaney, M. (2023). Neural architecture search for energy-efficient always-on audio machine learning. Neural Computing and Applications, 35, 12133-12144. https://doi.org/10.1007/s00521-023-08345-y

Saha, S., Xu, L., & Colleagues. (2025). Vision transformers on the edge: A comprehensive survey of model compression and acceleration strategies. ACM Computing Surveys. Advance online publication. https://arxiv.org/abs/2503.02891

Wang, A., Chen, H., Lin, Z., Han, J., & Ding, G. (2023). RepViT: Revisiting mobile CNN from ViT perspective. In Proceedings of CVPR 2024 (pp. 20985-20995). https://doi.org/10.48550/arXiv.2307.09283

Gao, S., Zhang, Y., Huang, F., & Huang, H. (2024). BilevelPruning: Unified dynamic and static channel pruning for convolutional neural networks. In Proceedings of CVPR 2024 (pp. 4471-4481). https://openaccess.thecvf.com/content/CVPR2024/papers/Gao_BilevelPruning_Unified_Dynamic_and_Static_Channel_Pruning_for_Convolutional_Neural_CVPR_2024_paper.pdf

Kim, N. J., Lee, J., & Kim, H. (2024). HyQ: Hardware-friendly post-training quantization for CNN-transformer hybrid networks. In Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI-24) (pp. 4291-4298). https://doi.org/10.24963/ijcai.2024/474

Luqman, A., Qazi, K., & Khan, I. (2024). Post-training non-uniform quantization for convolutional neural networks. Proceedings of the 41st International Conference on Machine Learning (ICML 2024). https://arxiv.org/abs/2412.07391

Lin, J., Chen, W.-M., Lin, Y., Cohn, J., Gan, C., & Han, S. (2021). MCUNet: Tiny deep learning on IoT devices. Advances in Neural Information Processing Systems, 33, 11734-11745. https://arxiv.org/abs/2007.10319

ISSN: 3049-3552

Volume 3 Issue 6, November-December 2025, pp 168-184

Liu, S., Zhou, W., Zhou, Z., Guo, B., Wang, M., Fang, C., Lin, Z., & Yu, Z. (2024). Deep learning inference on heterogeneous mobile processors: Potentials and pitfalls. arXiv Preprint. https://arxiv.org/abs/2405.01851

