International Journal of Computers and Informatics (IJCI)
Vol. (5), No. (11)
IJCI
November 2026
المجلة الدولية للحاسبات والمعلوماتية
الإصدار (5)، العدد (11)

Comparative Evaluation Methodologies for Algorithm Acceleration on Embedded Platforms: A Structured Review and Reproducible Assessment Framework

Mays Q. Sedeeq
Computer Technology Engineering Department, Technical Engineering College / Mosul, Northern Technical University, Iraq
ORCID: https://orcid.org/0009-0001-4625-9486
mays.q.sedee@ntu.edu.iq

Abstract

Background: Performance results for algorithms accelerated on microcontrollers, field-programmable gate arrays (FPGAs), multiprocessor systems-on-chip (MPSoCs), embedded GPUs, and application-specific accelerators are often reported under different workloads, accuracy requirements, and measurement conditions. Objective: This review examines how published studies measure execution time, throughput, energy, memory use, and hardware resources, and proposes a consistent approach to interpreting comparisons across platforms. Methods: A targeted review of publisher records, research repositories, and benchmarking standards identified 15 representative implementations published between 2015 and 2026. The analysis draws on 52 references, including primary studies, surveys, and methodological sources. Study outcomes were recorded with their associated tasks, platforms, measurement conditions, and level of source verification. Because the search was not an exhaustive database-level systematic search, the article is presented as a structured review rather than a PRISMA-compliant systematic review. Results: Reported outcomes vary substantially in both units and experimental scope. One FPGA CNN implementation reported 61.62 GFLOPS at 100 MHz; FINN reported 21,906 image classifications per second under a stated system power of less than 25 W; and a 2026 microcontroller study reported 38 ms execution time, 387 µJ per inference, and 174 kB of SRAM in a specified configuration. These results cannot be ranked directly without matching workload and measurement conditions. Discussion: Differences in power boundaries, memory definitions, hardware configurations, and baseline optimization limit cross-study interpretation. The proposed Cross-Platform Accelerator Evaluation Protocol (CAEP) provides a sequence of checks for workload equivalence, measurement boundaries, and reporting completeness before numerical comparisons are made. Conclusion: A defensible comparison begins by establishing whether results measure equivalent tasks under sufficiently similar conditions. The proposed protocol requires prospective validation, and a comprehensive systematic review would require a fully documented database search and independent full-text assessment.

Keywords: Algorithm Acceleration, Embedded Computing, FPGA, Mpsoc, Tinyml, Energy Efficiency, Memory Utilization, Performance Benchmarking, Reproducibility, Hardware Evaluation.

1. Introduction

Embedded systems increasingly perform time-sensitive signal processing, image analysis, machine-learning inference, and control tasks under strict constraints on energy consumption, memory capacity, and temperature. Selecting between microcontrollers, FPGAs, heterogeneous CPU–FPGA systems, and dedicated accelerators requires consideration of more than computational throughput. Data transfer, memory bandwidth, software overhead, and application-level timing can determine overall performance. The Roofline model established the importance of bandwidth limits [3], while subsequent studies showed how memory access and data reuse influence neural-network efficiency [8,17,18]. In embedded deployments, deadlines, operating duty cycles, and standby consumption introduce additional requirements that are often outside the scope of accelerator-level benchmarks [4–7,23,24].

Reported performance improvements are difficult to compare when studies use different definitions of workload, timing, or power consumption. Peak GFLOPS for a convolution kernel, for example, cannot be compared directly with end-to-end inference latency for another model. Similarly, chip-level power and complete-system power represent different measurement boundaries. Resource counts, including look-up tables (LUTs), registers, digital signal processing (DSP) blocks, and logic gates, are not directly comparable across FPGA and ASIC technologies [9–19,27–30]. Numerical precision and model compression can also alter accuracy and resource requirements [46–49]. Although recent surveys cover edge accelerators, FPGA design, heterogeneous computing, and transformer inference [31–39], comparative evaluation requires explicit rules for deciding when results are sufficiently comparable.

Earlier research by Sedeeq and colleagues examined embedded multiprocessor design and workload balancing on FPGAs. Their Mandelbrot implementation reported more than 20-fold performance improvement relative to the single-processor baseline used in that study [41,42]. Related work on FPGA-based laser-jamming systems explored spatial parallelism [43,44]. These studies illustrate important principles of concurrency and task-specific hardware design; however, their results cannot be treated as directly comparable with CNN or large-language-model inference. The present review considers these earlier contributions alongside contemporary benchmarking practices [5–7,25,26,40,45].

1.1. Research questions:

Four questions guide this review. RQ1: Which performance metrics support meaningful comparisons across embedded platforms? RQ2: How do studies define and measure energy consumption, memory use, bandwidth, and hardware resources? RQ3: Which reporting limitations reduce the reliability and reproducibility of published comparisons? RQ4: What evaluation procedure can support comparisons within equivalent workloads while accounting for differences between hardware technologies?

1.2. Contribution and scope:

This review provides four contributions. First, it compares the reported methods and results of 15 representative hardware implementations. Second, it records which details could be verified in the sources consulted, without assuming that unverified details were absent from the original publications. Third, it examines common sources of error in comparisons involving workload, timing, power, memory, and implementation resources. Finally, it proposes the Cross-Platform Accelerator Evaluation Protocol (CAEP), illustrated through published case studies. The proposed contribution lies in combining existing measurement principles into a practical comparison procedure, rather than introducing new performance metrics [1–8,20–26,50–52].

2. Review Method

2.1. Review design and reporting approach:

This study uses a structured critical review approach. Studies were selected to cover a range of embedded hardware technologies and evaluation practices rather than through an exhaustive search of indexed databases. A complete screening log, duplicate-record counts, full-text assessment for every potentially eligible record, and independent screening by multiple reviewers were not available. Accordingly, the study does not claim full compliance with the reporting requirements of a systematic review. PRISMA 2020 [1], PRISMA-S [2], and established classifications of review methodologies [51] were used to guide transparent reporting and to identify limitations. The literature cutoff was 24 October 2026; coverage of every publication available by that date is not claimed.

2.2. Sources, search approach, and study selection:

Relevant publications were identified through targeted searches of publisher websites, scholarly repositories, conference proceedings, and benchmark documentation, including sources associated with IEEE, ACM, NeurIPS, MLSys, USENIX, and MLCommons. Citation searching and established publications provided background studies, while work from 2024–2026 was included to reflect recent applications and benchmarking developments. Appendix A provides search expressions and selection procedures that can be reproduced in a subsequent comprehensive review. The original search did not produce a complete exportable set of records; therefore, search totals, deduplication counts, and a PRISMA screening flow are not reported.

Table (1). Eligibility criteria for the detailed comparative analysis.

Criterion	Inclusion	Exclusion or qualification
Hardware implementation	Physical or synthesizable embedded/accelerator implementation with platform identified	Simulation-only work without an identified hardware target, except methodological background
Algorithm & input	Identifiable task, model or signal/image workload	Claims that do not identify the task or workload
Evaluation	At least one explicitly reported timing, throughput, energy, memory or resource outcome	Qualitative performance claims without verifiable measurements

Criterion	Inclusion	Exclusion or qualification
Publication status	Publisher article, conference paper, scholarly institutional source	Unsubstantiated promotional claims
Historical scope	Primarily 2015–2026; older foundational work for context	Earlier methodology papers are not counted as recent hardware experiments
Duplication	Prefer archival journal or peer-reviewed conference version	Do not count preprint and journal version as separate implementations
Data traceability	Record exact publication URL/DOI and evidence-verification tier	Do not infer power, LUT, BRAM, or memory values that were not verified

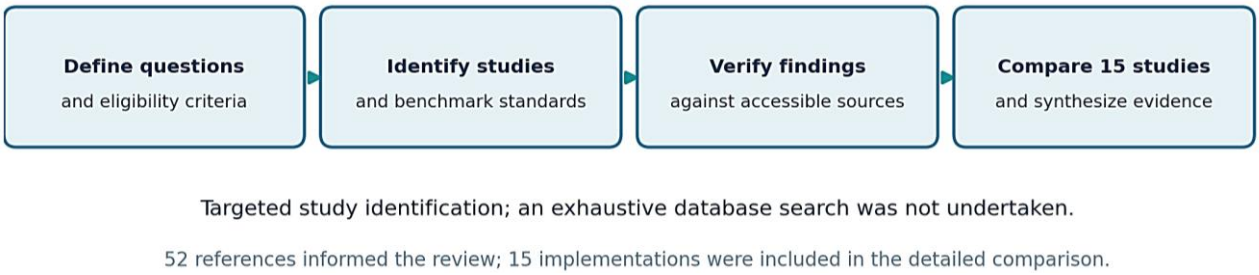


Figure (1). Literature identification and evidence synthesis used in this review. The process does not represent a complete PRISMA screening flow.

2.3. Data extraction and assessment of evidence:

For each selected study, information was collected on the computational task, dataset, platform, device and clock configuration, numerical precision, batch size or degree of parallelism, execution time, throughput, power or energy boundary, memory use, and hardware resources. Quantitative values were included only when supported by accessible primary-source material. The designation "NV" means that the specified detail could not be verified in the material consulted; it does not establish that the original publication failed to report it. "N/A" indicates that a measure does not apply to the hardware category, such as FPGA LUT use for a custom ASIC. Evidence was drawn from accessible research articles, author manuscripts, abstracts, and publisher records [12,17,18,20,21,27,28,30,41].

The assessment considered both the availability of reported information and potential sources of bias in performance comparisons. The latter include differences in workload or numerical precision, unspecified power-measurement boundaries, incomplete memory accounting, unequal compiler optimization, and limited reporting of variability or thermal conditions. The assessment identifies possible methodological concerns but does not constitute an independent, full-text risk-of-bias evaluation by two reviewers.

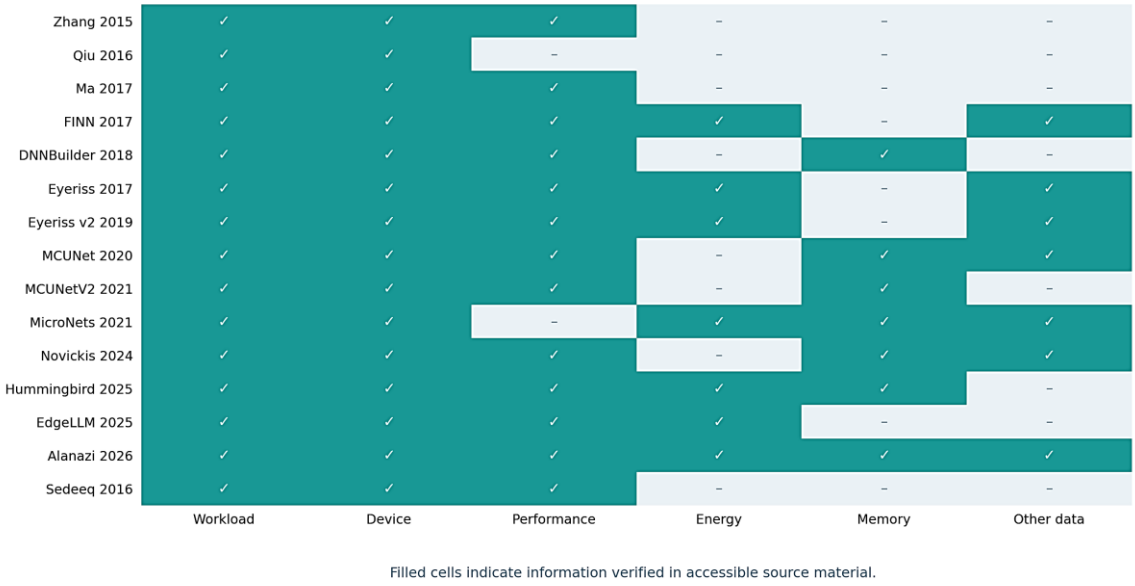


Figure (2). Information confirmed in the consulted material for each primary study. Unmarked entries indicate information not verified here, rather than information necessarily omitted by the original authors.

2.4. Evidence synthesis:

A pooled estimate of acceleration was not calculated because the selected studies evaluate different tasks, models, input sizes, devices, numerical formats, and baselines. Combining such measurements would yield a misleading summary. Instead, the results are organized by hardware category and analyzed according to the comparability of their experimental conditions. The synthesis distinguishes observed or author-reported measurements from interpretation based on the review framework.

3. Hardware Platforms and Measurement Criteria

The reviewed workloads include streaming signal and image processing, FPGA-based convolutional neural networks, quantized inference, memory-constrained microcontroller applications, ASIC-based neural processing, multiprocessor scheduling, and language-model inference on FPGA or CPU–FPGA platforms. Their performance depends on different patterns of data reuse, synchronization, memory traffic, and computational precision. Consequently, no single throughput metric adequately characterizes every application [8–30,40–45].

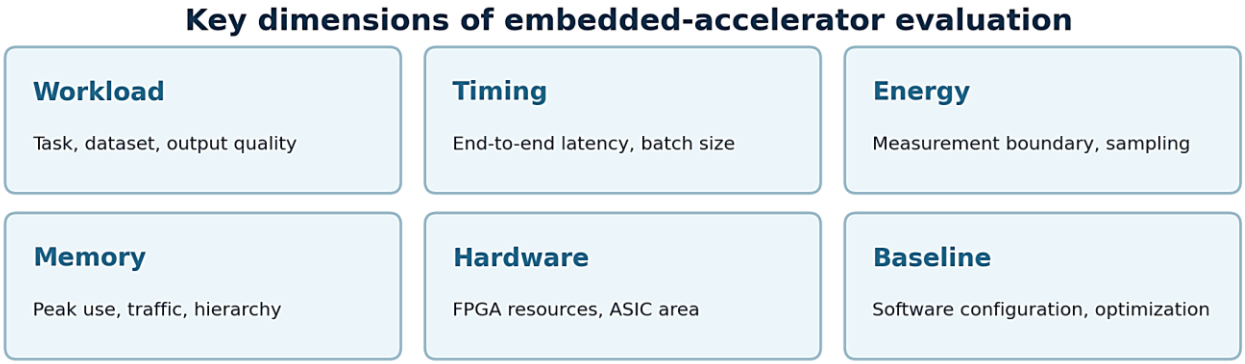


Figure (3). Six dimensions that should be documented when comparing acceleration results across embedded platforms.

Table (2). Relevant evaluation metrics and principal limitations by platform type.

Family	Recommended measures	Common limitation
Microcontroller	p50/p95 latency; µJ/inference; peak SRAM; flash; quality	Peak activation memory and sleep mode omitted
FPGA / SoC-FPGA	End-to-end and kernel time; frames/s; fabric resources; board W; DMA	Host transfers, BRAM and precision hidden
MPSoC	Throughput; deadline miss; per-core utilization; synchronization	Single-core baseline and load distribution mismatched
Embedded GPU	Kernel and application latency; thermal clock; input transfers	DVFS, kernel warmup and host boundary
ASIC / NPU	Inferences/J; die area; SRAM hierarchy; off-chip traffic	Chip power contrasted with full-board power
Hybrid CPU–FPGA	Token throughput; wall/rail energy; host/fabric shares	Model partition, DRAM and PCIe omitted

4. Comparative Analysis of Fifteen Studies

Tables (3) and (4) summarize quantitative findings together with the scope and limitations of the available supporting evidence. The reported results should not be interpreted as a ranking of hardware platforms. Values unavailable in the consulted sources are marked “NV” pending additional full-text verification. Claims such as “up to” or improvements over a study-specific baseline apply only to the conditions reported by the original authors.

Table (3). Platform and performance characteristics reported in 15 primary studies. NV: not verified in the sources consulted.

Study	Target and workload	Reported performance	Power or energy	Memory and hardware resources
Zhang et al. [9]	FPGA VC707; CNN	61.62 GFLOPS peak; 100 MHz	NV	NV
Qiu et al. [10]	Embedded FPGA; CNN	NV	NV	NV
Ma et al. [11]	Arria 10; VGG-16	645.25 GOPS; 47.97 ms	NV	NV
FINN [12]	ZC706; binarized CNN	21,906 images/s; 283 µs (CIFAR-10/SVHN)	<25 W system	NV
DNNBuilder [15]	XC7Z045/KU115; CNN/YOLO	Up to 4,218 GOPS; 7.7× latency reduction	NV	43× BRAM relative reduction
Eyeriss [17]	65 nm ASIC; AlexNet	35 frames/s; batch 4	278 mW (reported operating mode)	168 PEs
Eyeriss v2 [18]	65 nm ASIC; sparse MobileNet	1,470.6 inferences/s; batch 1	2,560.3 inferences/J	FPGA resources N/A
MCUNet [20]	MCU; image/audio tasks	1.7–3.3× vs cited TFLM/CMSIS-NN baselines	NV	3.4× less memory in stated comparison
MCUNetV2 [21]	STM32F746; TinyML CNNs	No universal latency figure	NV	4–6× measured peak SRAM reduction on MCU
MicroNets [22]	Commodity MCU; three TinyML tasks	Task-specific; NV in public abstract	Energy measured, raw values NV	Peak data dependent
Novickis et al. [27]	Kintex KU115; 2D image transform	275 frames/s; 512×512 use case	Power: tool estimate, not board W	51 DSP; 5,203 logic cells; 1 MB buffer
Hummingbird [28]	KV260 / ZCU104; LLaMA3-8B	4.8 / 8.6 tokens/s (different boards)	42% reported power saving vs named prior work	67% LUT, 39% DSP savings relative
EdgeLLM [29]	CPU–FPGA VCU128; LLM	1.91× throughput vs cited A100	7.55× energy efficiency vs cited A100	NV
Alanazi et al. [30]	STM32H743ZI; MLPerf Tiny task set	38 ms deployment (reported configuration)	387 µJ/inference	174 kB SRAM; 198 kB flash
Sedeeq et al. [41]	Nios II FPGA MPSoC; Mandelbrot	>20× versus paper single-core baseline	NV	NV

Table (4). Evidence sources and limitations affecting the interpretation of each study.

Source	Evidence consulted	Interpretation limit
Zhang [9]	Publisher abstract, explicit peak	Do not infer system power or application-level latency
Qiu [10]	Bibliography/accessible description	Missing energy and LUT values remain unverified
Ma [11]	Academic publisher-affiliated metadata	Different VGG workloads prevent a direct GOPS comparison
FINN [12]	Author-hosted primary paper; explicit case	System power is not chip-only power; precision also differs
DNNBuilder [15]	Official IEEE abstract and comparison claims	Peak claims may come from different tested workloads
Eyeriss [17]	IEEE abstract with power, batch, PEs	278 mW is not a wall-power measurement
Eyeriss v2 [18]	IEEE abstract with batch-1 result	Model and batch differences prevent a direct speedup estimate
MCUNet [20]	Research abstract with runtime ratios	Do not generalize runtime ratios to all tasks
MCUNetV2 [21]	Accessible paper: simulated and measured memory	Simulated and measured memory reductions have different definitions
MicroNets [22]	MLSys abstract; task conditions	Exact energy per inference was not verified in the available abstract
Novickis [27]	Open article; device, DSP, buffering	Tool-based power estimates are not direct measurements
Hummingbird [28]	IEEE abstract; device-specific token rate	Different boards do not provide a same-device speedup
EdgeLLM [29]	IEEE abstract; named A100 comparison	The stated A100 comparison does not apply to all GPUs
Alanazi [30]	Open article; MCU, energy, SRAM	The reported configuration must not be generalized to every task
Sedeeq [41]	Publisher archive and abstract	Mandelbrot speedup is not a neural-inference result

4.1. FPGA accelerators for convolutional neural networks:

Zhang et al. [9] used the relationship between computational capability and memory bandwidth to guide FPGA accelerator design. Their reported 61.62 GFLOPS is a peak computational result at 100 MHz on a VC707 platform, not a distribution of application-level latency. Ma et al. [11] reported 645.25 GOPS and a VGG-16 latency of 47.97 ms on an Arria 10 platform. Because the workloads, devices, and numerical configurations differ, these figures do not establish a direct speedup between the studies. DNNBuilder [15] used automated design exploration and reported up to a 43-fold reduction in BRAM use against the baseline specified by its authors; this does not represent a general reduction across all FPGA designs. FINN [12] showed the benefits of binary neural-network inference, while also illustrating the importance of workload and accuracy requirements. Together, these studies show why arithmetic throughput alone is insufficient for evaluating complete embedded applications [9–16,45].

4.2. ASIC and microcontroller accelerators:

The Eyeriss accelerator [17] reported 35 AlexNet frames per second and 278 mW at a batch size of four, emphasizing data reuse and reduced off-chip transfers. Eyeriss v2 [18] reported 1,470.6 sparse-MobileNet inferences per second and 2,560.3 inferences per joule at batch size one. These results concern different models and operating conditions and should not be combined into a general architecture-level performance ratio. MCUNet [20] reported 3.4-fold lower memory use and 1.7–3.3-fold faster execution than the baselines in its specified comparisons. MCUNetV2 [21] reported approximately 3.7–8-fold reductions in estimated peak memory and approximately 4–6-fold reductions in measurements on microcontroller hardware. Estimated allocation and measured device memory should be distinguished. MicroNets [22], CMSIS-NN [23], TensorFlow Lite Micro [24], and compiler systems [25,26] further illustrate how software libraries and optimization choices affect comparisons attributed to hardware.

4.3. Recent task-specific accelerators:

A 2024 FPGA image-transformation accelerator [27] reported 275 frames per second for a 512 × 512 rotation task, using 51 DSP blocks and a 1-MB buffer. Its reported power estimates were obtained from design tools rather than direct board measurements. Hummingbird [28] reported 4.8 tokens per second on a KV260 platform and 8.6 tokens per second on a ZCU104 platform for LLaMA3-8B; the two figures describe different devices. EdgeLLM [29] reported throughput and energy-efficiency improvements relative to a particular A100-based configuration, but the comparison does not establish superiority over all GPU systems. A 2026 Cortex-M7 implementation [30] reported execution time, task energy, and SRAM usage for a specified setup, providing an example of multidimensional reporting. Recent surveys [31–39] show that changes in compiler systems, memory architectures, and model complexity continue to complicate evaluation.

5. Assessment of Reporting Quality and Evidence Limitations

5.1. Scope of the assessment:

Figure (2) records whether selected details could be confirmed in the material consulted for each implementation. An unmarked entry does not mean that the original publication omitted the information. A complete risk-of-bias assessment would require the full text of every eligible study and independent consideration of measurement instrumentation, baseline selection, and reported outcomes. Table (5) provides a consistent set of criteria for such an assessment.

Table (5). Methodological criteria recommended for full-text quality assessment.

Domain	Assessment question	Supporting evidence
C1 Workload equivalence	Are identical input sets, model versions and correctness targets applied?	Data manifest; task-specific accuracy/error tables
C2 Timing	Is end-to-end execution distinguished from accelerator-kernel time?	Timing diagrams; synchronization and p50/p95
C3 Power	What physical and temporal boundary was measured?	Rail list, sensor, sampling, idle correction, duty cycle
C4 Memory	Is peak SRAM vs flash vs DRAM traffic unambiguously separated?	Profiling tool and measured peak/allocation trace
C5 FPGA fabric	Are LUT, FF, DSP, BRAM and tool/clock configurations identified?	Final place-and-route report and target device
C6 Baseline effort	Are compilation and optimization efforts comparably tuned?	Baselines, flags, libraries and platform revisions
C7 Replication	Are runs/seeds, variability, software versions and artifacts given?	Sample count, intervals, source and measurement scripts
C8 Generality	Are conclusions limited to tasks, models and hardware actually evaluated?	Ablations, multiple model families, sensitivity analysis

5.2. Strength and interpretation of the evidence:

Detailed research articles generally allow closer examination of methods and results than abstracts alone. For this reason, the review identifies the type of source supporting each extracted finding and treats numerical outcomes as author-reported unless independently measured. A study is not classified as methodologically weak merely because its complete text was unavailable to this review. Formal certainty ratings and meta-analytic estimates were not produced. Future work should apply the criteria in Table (5) through independent full-text assessments by at least two reviewers, with disagreements resolved and documented.

6. Challenges in Comparing Embedded Accelerator Performance

6.1. Workload equivalence and numerical precision:

Meaningful throughput comparisons require equivalent tasks and output-quality requirements. Quantization, sparse computation, operator fusion, and changes in model architecture can alter the number and type of operations performed. Networks using binary, 4-bit, or 16-bit arithmetic are therefore not equivalent solely because they process the same dataset. These choices also affect predictive accuracy [12–14,20–22,46–49]. Performance claims should be considered together with accuracy and error tolerance. A faster but less accurate implementation may represent a useful trade-off, but it does not establish an unconditional advantage over another system.

6.2. Power-measurement boundaries and operating cycles:

Average power, expressed in watts, is not the same as energy per task, expressed in joules. Over a defined interval, task energy is the integral of power with respect to time. Dynamic energy after subtracting idle consumption may also be reported, provided the idle reference is defined. Eyeriss

reports an ASIC operating point of 278 mW [17], whereas FINN reports total-system power below 25 W for its ZC706 implementation [12]. These measurements cover different components and cannot be used directly to rank energy efficiency. MLPerf Tiny provides common tasks and power-measurement procedures [5,6,50], and MLPerf Power addresses standardized efficiency reporting [7]. For intermittently operated devices, measurements should include any relevant acquisition, wake-up, computation, and idle intervals, with power rails and sampling procedures specified.

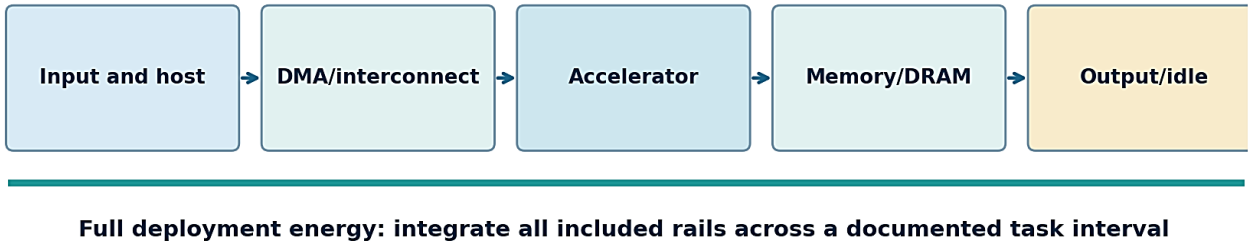


Figure (4). Components that may contribute to total task energy. The diagram is illustrative and contains no measured power data.

6.3. Memory capacity, data movement, and execution time:

Memory requirements include model storage in flash or ROM, temporary activation storage in SRAM or BRAM, transfer buffers, and data movement to and from external DRAM. DNNBuilder's reported 43-fold BRAM reduction [15] and MCUNetV2's approximately 4–6-fold reduction in measured peak SRAM [21] cannot be compared directly because they involve different memory types, workloads, and baselines. Larger local buffers can also increase throughput by supporting parallel access, as illustrated in the FPGA image-transformation study [27]. Language-model inference introduces additional constraints associated with model weights, memory bandwidth, and key–value caches [28,29,36]. Evaluations should therefore report both memory capacity and data traffic.

6.4. Hardware resource utilization and baseline optimization:

FPGA utilization depends on the specific device and implementation constraints. Counts of LUTs and DSP blocks cannot be converted directly into ASIC area or processing-element counts. Ratios such as GOPS per DSP may assist comparisons within similar FPGA families, but only when precision, clock frequency, memory use, and operating conditions are reported. These ratios can also be misleading if an implementation moves arithmetic from DSP blocks into LUT fabric. Compiler choice and optimization effort affect achievable performance [15,16,25,26]; comparisons with minimally optimized CPU or GPU software may overstate acceleration benefits. Reported speedups should therefore be interpreted against the precise baselines used in each study [9–11,45].

6.5. Parallel execution and non-neural workloads:

Parallel performance depends on the proportion of serial work, synchronization overhead, memory contention, and uneven distribution of tasks. Amdahl's law [52] describes the theoretical effect of serial execution on speedup, but realistic heterogeneous systems also require consideration of communication and scheduling costs. Sedeeq and colleagues [41,42] investigated FPGA-based multiprocessors and load balancing. The FPGA laser-jamming studies [43,44] examined spatial parallelism in specialized signal-processing architectures. These examples extend the discussion beyond neural-network inference and reinforce the need for workload-specific correctness and timing criteria.

6.6. Uncertainty, reproducibility, and selection bias:

A peak result or an "up to" improvement does not establish consistent performance in real-time operation. Where practical, experiments should report repeated measurements, median and tail latency, clock settings, operating temperature, variance, and uncertainty intervals. Published results may emphasize favorable configurations, while the targeted study selection used here can also introduce selection bias. Differences in measurement procedures and incomplete source access prevent a reliable pooled effect estimate for these 15 implementations. Public implementation artifacts, synthesis reports, power traces, and recorded compiler versions would improve reproducibility [4–7,23–26,45,50].

7. Cross-Platform Accelerator Evaluation Protocol (CAEP)

The Cross-Platform Accelerator Evaluation Protocol (CAEP) is proposed as a practical procedure for assessing whether published acceleration results can be compared. It complements, rather than replaces, established benchmarking standards. The procedure first checks task and output-quality equivalence, then establishes compatible timing and power boundaries, documents memory and hardware measures, and records uncertainty and baseline conditions. A direct numerical comparison is recommended only when these requirements are sufficiently matched. The protocol has not yet been independently validated across a common experimental benchmark suite.

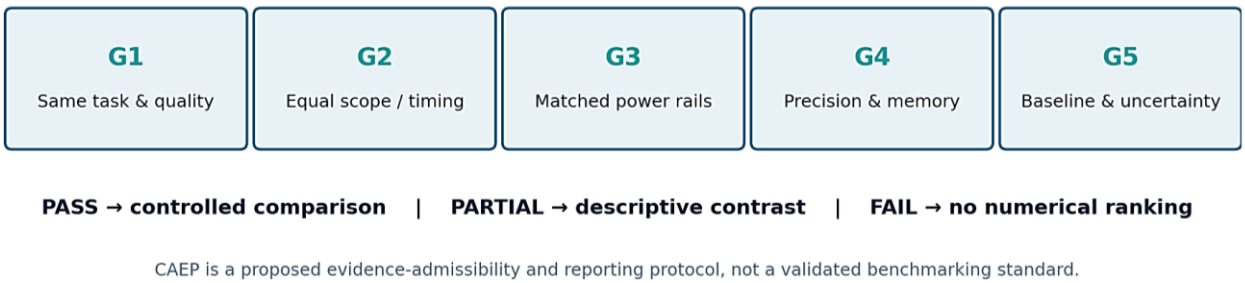


Figure (5). Proposed CAEP procedure for assessing whether a quantitative comparison is justified.

Table (6). CAEP criteria, documentation requirements, and permitted interpretations.

Gate	Condition for comparison	Required documentation
G1: Task	Identical input task/data, accepted output tolerance	Manifest, preprocessing, precision, accuracy/error
G2: Timing	Same end-to-end definition and concurrency/batch	Timing boundary, repeated latency and throughput
G3: Energy	Identical declared rail + comparable task interval	Power trace, sampling rate, idle/state policy
G4: Memory/hardware	Same definition for compared memory or resources	Peak allocation, off-chip traffic, exact FPGA part
G5: Baseline/variation	Matching software effort and repeated runs	Tool versions, clock, seeds, thermal/DVFS, intervals
Decision A: Controlled	All relevant gates pass	Direct ratio/percentage change allowed
Decision B: Descriptive	Some conditions unavailable or unmatched	Only side-by-side reported values; no cross-study ranking
Decision C: Inadmissible	Different task/accuracy or incompatible denominator	Do not compute performance advantage

7.1. Recommended measurements and reporting:

For each task and hardware configuration, evaluations should report median and 95th-percentile latency, throughput in appropriate task-specific units, task energy and its measurement boundary, peak SRAM and flash use, off-chip transfers, and sustained memory bandwidth. FPGA studies should additionally report the exact device, achieved clock frequency, and utilization of LUTs, flip-flops, DSP blocks, BRAM, and URAM, as applicable. Output quality, missed deadlines, and uncertainty intervals should accompany performance measures where supported by repeated trials. Pareto analysis is appropriate only for comparisons involving equivalent tasks and compatible easurements. CAEP does not combine all outcomes into a single universal score.

7.2. Illustrative applications of CAEP:

Table (7). Illustrative CAEP assessments based on results reported in the cited literature.

Comparison	Assessment	Appropriate interpretation
Zhang [9] 61.62 GFLOPS vs Ma [11] 645.25 GOPS	G1, G2, G4 not established across studies	Describe architecture/design claims; do not compute 10.47× accelerator superiority
Eyeriss [17] 278 mW vs FINN [12] <25 W system	G1 and G3 fail (task and energy scope)	No power-efficiency ranking
Eyeriss [17] vs v2 [18] as cross-paper headline	Model/batch/precision and scope differ	Do not compare AlexNet power to sparse MobileNet inference/J
MCUNetV2 [21] 4–6× MCU SRAM vs DNNBuilder [15] 43× BRAM	G4 fails (memory technology/denominator)	Both are local memory optimizations; effects not commensurate

Comparison	Assessment	Appropriate interpretation
Hummingbird [28] 4.8 and 8.6 tokens/s	G1 same model; G4 target board differs	Report board-specific deployment rates; not an architectural speedup
Alanazi [30] 38 ms & 387 µJ vs any unmeasured MCU study	G3 unestablished for comparator	Single-study multi-objective example; no cross-study Pareto front

7.3. Evidence and future validation:

The reported measurements in Table (7) are drawn from the cited publications, while the assessment of comparability is the interpretation developed in this review. No additional performance experiments, power measurements, or FPGA implementations were conducted. CAEP should be tested in a prospective study that applies the same benchmark tasks to at least one microcontroller, one FPGA, and one heterogeneous CPU–FPGA platform. That evaluation should include calibrated system-level power measurements, matched accuracy criteria, and independent replication.

8. Research Gaps and Directions for Validation

Several unresolved issues emerge from the comparative analysis. First, studies measure energy at different levels, including chip, device, board, and complete deployment [5–7,17,27,30]. Second, memory-capacity results often omit the volume and cost of external data transfers [9–11,15,18,20,21,28,29]. Third, compiler settings, batching, precision, and optimization effort may differ between an accelerator and its software baseline [23–26,45]. Fourth, reporting frequently emphasizes throughput without equivalent attention to output quality, deadlines, or measurement uncertainty [12,14,22,27,30]. Finally, comparative reviews often focus on neural networks, even though image processing, signal processing, and FPGA multiprocessors raise similar methodological questions [40–44].

Future validation should use published benchmark definitions for convolution, image interpolation, and a streaming workload; establish consistent timing markers from input acquisition to valid output; and collect calibrated power traces under documented operating conditions. It should also record peak SRAM, off-chip data traffic, final routed FPGA resources, and baseline compiler settings. Independent teams could then repeat selected experiments to determine whether CAEP improves agreement between evaluations and changes conclusions derived from single-metric rankings. These activities are recommendations for future research, not experiments reported in this article.

9. Limitations

The 15 implementations were selected to illustrate methodological differences rather than identified through an exhaustive database search, so selection bias remains possible. Some findings were checked against accessible full texts, whereas others were available only through abstracts or publisher records. The 52 references include primary studies, surveys, standards, and methodological sources; they are not 52 independently assessed accelerator experiments. The evidence matrix records what could be confirmed from the available material and should not be interpreted as a complete

evaluation of study quality. Independent duplicate screening, formal full-text risk-of-bias assessment, and statistical synthesis were not undertaken. The present article is therefore described as a structured review. A complete systematic review would require documented database searches, record screening, and consistent appraisal of all eligible full texts.

10. Conclusions

Evaluating embedded algorithm acceleration requires consideration of workload, accuracy, timing, energy, memory, and hardware constraints. The 15 studies reviewed here report useful experimental findings, but their performance measures often differ in units, device configurations, and system boundaries. Direct numerical rankings are therefore justified only when tasks and measurement conditions are comparable. This review proposes CAEP as a procedure for documenting these conditions and identifying when a comparison is appropriate. The framework offers a consistent method for interpreting the existing literature, but it does not constitute an experimentally validated improvement in accelerator performance. Further work should test the protocol with common benchmarks and independently reproduced measurements.

Declarations

Ethics approval: Not applicable; the study analyzes published literature. Funding: To be confirmed by the authors. Competing interests: To be confirmed by the authors. Data availability: The reported study comparisons and source links appear in Tables 3–4 and the reference list; the proposed extraction fields appear in Appendix B. Author contributions and acknowledgments: To be completed by the authors. No new accelerator experiments were performed.

Appendix A. Search Strategy for a Comprehensive Review

The following search expressions provide a reproducible starting point for a future comprehensive systematic review; they do not represent an exhaustive search already completed for this article. Query A (IEEE Xplore and ACM Digital Library): ("FPGA" OR "MPSoC" OR "system on chip" OR "embedded GPU" OR "microcontroller" OR "NPU") AND (accelerat* OR "hardware optimization" OR "parallel computation") AND (latency OR throughput OR energy OR power OR memory OR benchmark). Query B (2024–2026 studies): ("edge AI" OR TinyML OR "embedded inference" OR "FPGA LLM") AND (benchmark* OR power OR "energy efficiency" OR SRAM OR "resource utilization"). Query C (other FPGA workloads): ("spatial parallelism" OR "real-time image processing" OR Mandelbrot OR "spatial transformation") AND FPGA AND (performance OR resources). For the comprehensive extension, screen implementation studies from 2015–2026 while retaining earlier methodological papers as background. Export all bibliographic records; remove duplicates by DOI and title; screen titles and abstracts, followed by full texts; and document exclusion reasons and reviewer disagreements. Record the database, final query syntax, search date, and screening outcomes before preparing a PRISMA 2020 flow diagram.

Appendix B. Data Extraction and Reporting Checklist

Table (8). Recommended data fields and treatment of unavailable information.

Field	Mandatory subfields	If unavailable
Identity	DOI, publisher URL, year, version, artifact link	Mark “not verified in accessible material”
Task	Dataset, model, resolution/sequence, batch, precision, quality	Exclude from numerical cross-task ratio
Timing	Kernel/end-to-end, mean/median/p95, warmup, runs	Descriptive claim only
Power	Sensor, included components, rail/wall, interval, idle	Never infer joules from unmatched watts
Memory	Peak activation SRAM, flash, DRAM traffic, memory tool	Do not merge RAM/flash/BRAM
Hardware	Part number, clock, LUT/FF/DSP/BRAM/URAM or die area	No cross-technology normalized resource score
Baseline	System, quality, toolchain, optimization, DVFS	No generalizable speedup claim
Validity	Repeats, CI, workloads tested, operating temperature	Flag uncertainty and generalization boundary

A complete systematic review should document the eligibility criteria, databases, exact searches, record counts, screening decisions, full-text appraisal, synthesis methods, protocol registration status, funding, and competing interests [1,2,51]. In this review, the eligibility criteria and approach to evidence synthesis are described, and search expressions are provided for future use. Complete database export records, independent screening, and full-text appraisal for every eligible publication were not available; these procedures are not claimed as completed.

References

[1] Page, M. J., et al. (2021). The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. BMJ 372, n71. https://doi.org/10.1136/bmj.n71

[2] Rethlefsen, M. L., et al. (2021). PRISMA-S: an extension to the PRISMA statement for reporting literature searches in systematic reviews. Systematic Reviews 10, 39. https://doi.org/10.1186/s13643-020-01542-z

[3] Williams, S., Waterman, A., and Patterson, D. (2009). Roofline: an insightful visual performance model for multicore architectures. Communications of the ACM 52(4), 65-76. https://doi.org/10.1145/1498765.1498785

[4] Reddi, V. J., et al. (2020). MLPerf Inference Benchmark. Proceedings of ISCA 2020. https://doi.org/10.1109/ISCA45697.2020.00045

[5] Banbury, C., et al. (2021). MLPerf Tiny Benchmark. NeurIPS Datasets and Benchmarks Track. https://arxiv.org/abs/2106.07597

[6] MLCommons (2026). MLPerf Tiny v1.4 results and benchmark rules. MLCommons, 7 July 2026. https://mlcommons.org/2026/07/mlperf-tiny-v1-4-results/

[7] Tschand, A., Idgunji, S., Rajan, A. T. R., Kanter, D., and Reddi, V. J. (2026). MLPerf Power: Addressing the AI Power Wall. IEEE Micro, early access. https://doi.org/10.1109/MM.2026.3700126

[8] Sze, V., Chen, Y.-H., Yang, T.-J., and Emer, J. S. (2017). Efficient processing of deep neural networks: a tutorial and survey. Proceedings of the IEEE 105(12), 2295-2329.

https://doi.org/10.1109/JPROC.2017.2761740

[9] Zhang, C., Li, P., Sun, G., Guan, Y., Xiao, B., and Cong, J. (2015). Optimizing FPGA-based accelerator design for deep convolutional neural networks. ACM/SIGDA International Symposium on FPGAs. https://www.microsoft.com/en-us/research/publication/optimizing-fpga-based-accelerator-design-for-deep-convolutional-neural-networks/

[10] Qiu, J., et al. (2016). Going deeper with embedded FPGA platform for convolutional neural network. ACM/SIGDA International Symposium on FPGAs, 26-35.

https://doi.org/10.1145/2847263.2847265

[11] Ma, Y., Cao, Y., Vrudhula, S., and Seo, J.-S. (2017). Optimizing loop operation and dataflow in FPGA acceleration of deep convolutional neural networks. ACM/SIGDA International Symposium on FPGAs, 45-54. https://doi.org/10.1145/3020078.3021736

[12] Umuroglu, Y., et al. (2017). FINN: a framework for fast, scalable binarized neural network inference. ACM/SIGDA International Symposium on FPGAs. https://arxiv.org/abs/1612.07119

[13] Blott, M., Preusser, T. B., Fraser, N. J., Gambardella, G., O'Brien, K., and Umuroglu, Y. (2018). FINN-R: an end-to-end deep-learning framework for fast exploration of quantized neural networks. ACM Transactions on Reconfigurable Technology and Systems / preprint.

https://arxiv.org/abs/1809.04570

[14] Duarte, J., et al. (2018). Fast inference of deep neural networks in FPGAs for particle physics. Journal of Instrumentation 13, P07027. https://doi.org/10.1088/1748-0221/13/07/P07027

[15] Zhang, X., et al. (2018). DNNBuilder: an automated tool for building high-performance DNN hardware accelerators for FPGAs. ICCAD 2018, Article 56.

https://doi.org/10.1145/3240765.3240801

[16] Sharma, H., Park, J., Mahajan, D., Amaro, E., Kim, J. K., Shao, C., Mishra, A., and Esmaeilzadeh, H. (2016). From high-level deep neural models to FPGAs. 49th IEEE/ACM International Symposium on Microarchitecture. https://doi.org/10.1109/MICRO.2016.7783720

[17] Chen, Y.-H., Krishna, T., Emer, J. S., and Sze, V. (2017). Eyeriss: an energy-efficient reconfigurable accelerator for deep convolutional neural networks. IEEE Journal of Solid-State Circuits 52(1), 127-138. https://doi.org/10.1109/JSSC.2016.2616357

[18] Chen, Y.-H., Yang, T.-J., Emer, J. S., and Sze, V. (2019). Eyeriss v2: a flexible accelerator for emerging deep neural networks on mobile devices. IEEE Journal on Emerging and Selected Topics in Circuits and Systems 9(2), 292-308. https://doi.org/10.1109/JETCAS.2019.2910232

[19] Jouppi, N. P., et al. (2017). In-datacenter performance analysis of a tensor processing unit. ISCA 2017, 1-12. https://doi.org/10.1145/3140659.3080246

[20] Lin, J., Chen, W.-M., Lin, Y., Cohn, J., Gan, C., and Han, S. (2020). MCUNet: tiny deep learning on IoT devices. Advances in Neural Information Processing Systems 33. https://papers.neurips.cc/paper_files/paper/2020/hash/86c51678350f656dcc7f490a43946ee5-Abstract.html

[21] Lin, J., Chen, W.-M., Cai, H., Gan, C., and Han, S. (2021). MCUNetV2: memory-efficient patch-based inference for tiny deep learning. Advances in Neural Information Processing Systems 34. https://papers.nips.cc/paper_files/paper/2021/hash/1371bccec2447b5aa6d96d2a540fb401-Abstract.html

[22] Banbury, C., et al. (2021). MicroNets: neural network architectures for deploying TinyML applications on commodity microcontrollers. Proceedings of Machine Learning and Systems 3. https://proceedings.mlsys.org/paper_files/paper/2021/hash/c4d41d9619462c534b7b61d1f772385e-Abstract.html

[23] Lai, L., Suda, N., and Chandra, V. (2018). CMSIS-NN: efficient neural network kernels for Arm Cortex-M CPUs. arXiv:1801.06601 / Arm technical work. https://arxiv.org/abs/1801.06601

[24] David, R., et al. (2021). TensorFlow Lite Micro: embedded machine learning for TinyML systems. Proceedings of Machine Learning and Systems 3.

https://proceedings.mlsys.org/paper_files/paper/2021/hash/6c44dc73014d66ba49b28d483a8f8b0d-Abstract.html

[25] Chen, T., et al. (2018). TVM: an automated end-to-end optimizing compiler for deep learning. 13th USENIX OSDI. https://www.usenix.org/conference/osdi18/presentation/chen

[26] Zheng, L., et al. (2020). Ansor: generating high-performance tensor programs for deep learning. 14th USENIX OSDI. https://www.usenix.org/system/files/osdi20-zheng.pdf

[27] Novickis, R., Lielāmurs, E., Justs, D. J., Cvetkovs, A., and Ozols, K. (2024). Spatial transformation accelerator with parallel data access scheme for sample reconstruction. Electronics 13(5), 922. https://doi.org/10.3390/electronics13050922

[28] Li, J., Li, T., Chen, R., Shen, G., Zhao, D., Zhang, Q., and Zeng, Y. (2025). Hummingbird: a smaller and faster large language model accelerator on embedded FPGA. IEEE/ACM ICCAD 2025. https://doi.org/10.1109/ICCAD66269.2025.11241002

[29] Huang, M., Shen, A., Li, K., Peng, H., Li, B., and Yu, H. (2025). EdgeLLM: a highly efficient CPU-FPGA heterogeneous edge accelerator for large language models. IEEE Transactions on Circuits and Systems I 72(7), 3352-3365. https://doi.org/10.1109/TCSI.2025.3546256

[30] Alanazi, A. G., Alanazi, H. A., and Albalawi, N. S. (2026). Cross-layer resource optimization for ultra-low-power TinyML inference on ARM Cortex-M microcontrollers. Electronics 15(13), 2918. https://doi.org/10.3390/electronics15132918

[31] Hussein, Y. M., Hassan, R. F., and Chisab, R. F. (2026). FPGA-based reconfigurable SoCs for safety-critical AI inference: a systematic literature review. Electronics 15(12), 2695.

https://doi.org/10.3390/electronics15122695

[32] Cordova-Cardenas, R., Amor, D., and Gutiérrez, Á. (2025). Edge AI in practice: a survey and deployment framework for neural networks on embedded systems. Electronics 14(24), 4877. https://doi.org/10.3390/electronics14244877

[33] Alam, S., Yakopcic, C., Wu, Q., Barnell, M., Khan, S., and Taha, T. M. (2024). Survey of deep learning accelerators for edge and emerging computing. Electronics 13(15), 2988. https://doi.org/10.3390/electronics13152988

[34] Liu, F., Li, H., Hu, W., and He, Y. (2024). Review of neural network model acceleration techniques based on FPGA platforms. Neurocomputing 610, 128511.

https://doi.org/10.1016/j.neucom.2024.128511

[35] Zhang, Z. (2026). A review of embedded artificial intelligence research (2023-2026): technological advancements, representative advances, and future prospects. Micromachines 17(5), 586. https://doi.org/10.3390/mi17050586

[36] Blankestijn, A., Odyurt, U., and Yousefzadeh, A. (2026). Recent developments in Transformer inference deployment on FPGA platforms: a survey. Journal of Systems Architecture 177, 103841. https://doi.org/10.1016/j.sysarc.2026.103841

[37] Silvano, C., et al. (2025). A survey on deep learning hardware accelerators for heterogeneous HPC platforms. ACM Computing Surveys 57(11), Article 286. https://doi.org/10.1145/3729215

[38] Morsy, M. M. (2026). AI-on-chip systems: a cross-layer review of architectures, interconnects, design automation, and embedded intelligence. Electronics 15(12), 2645.

https://doi.org/10.3390/electronics15122645

[39] Huang, Y., Ando, T., Sebastian, A., Chang, M.-F., Yang, J. J., and Xia, Q. (2024). Memristor-based hardware accelerators for artificial intelligence. Nature Reviews Electrical Engineering 1, 286-299. https://doi.org/10.1038/s44287-024-00037-6

[40] Bieszczad, G. (2016). SoC-FPGA embedded system for real-time thermal image processing. MIXDES 2016, 469-473. https://doi.org/10.1109/MIXDES.2016.7529788

[41] Sedeeq, M. Q., Salih, M. H., Yousif, O. F., and Mohammed, N. Q. (2016). The vantage of utilizing FPGAs in the design of an embedded multiprocessor. ARPN Journal of Engineering and Applied Sciences 11(23), 13815-13827.

https://www.arpnjournals.com/jeas/volume_23_2016.htm

[42] Sedeeq, M. Q., Salih, M. H., Yousif, O. F., and Mohammed, N. Q. (2017). Insight on MPSoC environment and its performance related to load balance. ARPN Journal of Engineering and Applied Sciences 12(1). https://www.arpnjournals.com/jeas/volume_01_2017.htm

[43] Salih, M. H., Yousif, O. F., Seddeq, M. Q., and Isam, H. M. (2015). Enhance implementation of FPGA based laser missile frequency jamming system using spatial parallelism. ARPN Journal of Engineering and Applied Sciences 10(20), 9626-9632.

https://www.arpnjournals.org/jeas/research_papers/rp_2015/jeas_1115_2918.pdf

[44] Yousif, O. F., Salih, M. H., Hassnawi, L. A., Albreem, M. A. M., Seddeq, M. Q., and Isam, H. M. (2015). Design and implementation computing unit for laser jamming system using spatial parallelism on FPGA. IEEE ICSIPA 2015, 38-43. https://doi.org/10.1109/ICSIPA.2015.7412160

[45] Ducasse, Q., Cotret, P., Lagadec, L., and Stewart, R. (2021). Benchmarking quantized neural networks on FPGAs with FINN. arXiv:2102.01341. https://arxiv.org/abs/2102.01341

[46] Tan, M., and Le, Q. (2019). EfficientNet: rethinking model scaling for convolutional neural networks. Proceedings of ICML 2019. https://proceedings.mlr.press/v97/tan19a.html

[47] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). MobileNetV2: inverted residuals and linear bottlenecks. CVPR 2018.

https://openaccess.thecvf.com/content_cvpr_2018/html/Sandler_MobileNetV2_Inverted_Residuals_CVPR_2018_paper.html

[48] Han, S., Mao, H., and Dally, W. J. (2016). Deep compression: compressing deep neural networks with pruning, trained quantization and Huffman coding. International Conference on Learning Representations (ICLR). https://arxiv.org/abs/1510.00149

[49] Howard, A. G., et al. (2017). MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861. https://arxiv.org/abs/1704.04861

[50] MLCommons Tiny Working Group (2026). MLPerf Tiny: benchmarking AI at the edge, v1.4 release commentary. MLCommons official technical communication, 7 July 2026.

https://mlcommons.org/2026/07/mlperf-tiny-v1-4-results/

[51] Grant, M. J., and Booth, A. (2009). A typology of reviews: an analysis of 14 review types and associated methodologies. Health Information and Libraries Journal 26(2), 91-108.

https://doi.org/10.1111/j.1471-1842.2009.00848.x

[52] Amdahl, G. M. (1967). Validity of the single processor approach to achieving large scale computing capabilities. AFIPS Spring Joint Computer Conference.

https://doi.org/10.1145/1465482.1465560