EcoNAS: Carbon and Cost-Aware Neural Architecture Search for Edge Vision Applications

Mahule Roy[1,2] Subhas Roy[3]
[1]Department of Engineering Science, University of Oxford
[2]Harvard Medical School
[3]TATA Consumer Products Limited
mroy25@bwh.harvard.edu

Abstract

*Neural Architecture Search (NAS) has emerged as a powerful tool for automating the design of high-performance neural networks. While existing methods typically focus on optimizing accuracy or latency, practical considerations such as energy consumption and deployment cost remain underexplored, particularly in resource-constrained edge environments. In this work, we propose **EcoNAS**, a multi-objective NAS framework that jointly optimizes model accuracy, energy efficiency, and estimated deployment cost. EcoNAS employs a hardware-aware search space alongside training-free performance proxies, enabling rapid evaluation of candidate architectures and efficient exploration of the Pareto front without requiring full model training. We conduct extensive experiments on image classification and segmentation tasks across multiple edge platforms, including NVIDIA Jetson Nano, Raspberry Pi 4, and a simulated Edge TPU. Our results demonstrate that EcoNAS identifies architectures achieving competitive accuracy while substantially reducing energy consumption and deployment cost relative to standard NAS baselines and widely used manually designed networks. We further provide ablation studies examining the impact of proxy selection, search algorithm, and hardware modeling, alongside comprehensive implementation details to facilitate reproducibility.*

1. Introduction

The rapid adoption of deep learning on edge devices has created an increasing demand for models that are not only accurate but also resource-efficient and cost-effective for large-scale deployment. Neural Architecture Search (NAS) has emerged as a powerful tool for automating the design of high-performance neural networks, reducing reliance on manual architecture engineering. Traditional NAS approaches primarily focus on optimizing model accuracy under computational constraints such as latency or parameter count [1, 2]. However, these approaches often overlook critical practical considerations, including energy consumption, monetary cost, and carbon footprint, which are particularly relevant for sustainable edge AI applications. Inference across millions of deployed edge devices can contribute substantially to overall energy consumption and associated carbon emissions [3], while deployment costs—including hardware procurement, energy usage, and maintenance—pose significant real-world constraints. Existing energy-aware NAS methods typically target data-center GPUs and rely on simplified power models or latency as a proxy for energy [4], limiting their applicability to battery-powered edge devices with heterogeneous hardware and stricter energy budgets. Few prior works explicitly incorporate monetary cost into the search process, further restricting their relevance for practical deployment scenarios where operational expenditures are critical. To address these limitations, we propose **EcoNAS**, a multi-objective NAS framework designed for edge vision applications that jointly optimizes model accuracy, energy consumption, and estimated deployment cost under realistic hardware constraints, such as peak memory, latency, and parameter budgets. EcoNAS leverages a hardware-aware search space specifically tailored for edge devices and employs training-free performance proxies [5], enabling rapid evaluation of candidate architectures without requiring full model training. While our experiments target NVIDIA Jetson Nano, Raspberry Pi 4, and a simulated Coral Edge TPU, the EcoNAS framework is designed to be hardware-agnostic and can be extended to additional edge platforms, including smartphones and embedded MCUs, with minimal profiling effort. By efficiently exploring the Pareto-optimal front across multiple objectives, EcoNAS provides a practical methodology for selecting architectures that balance accuracy, efficiency, and cost. We validate EcoNAS through comprehensive experiments on both image classification

and semantic segmentation tasks across multiple edge platforms. Our results demonstrate that EcoNAS identifies architectures achieving competitive accuracy while substantially reducing energy consumption and deployment cost compared to standard NAS baselines and widely used manually designed networks. Furthermore, we conduct extensive ablation studies evaluating the impact of proxy selection, search algorithm, and hardware modeling. All implementation details, including code, search spaces, and hardware profiling scripts, are released to facilitate reproducibility and support future research on sustainable NAS for edge devices.

2. Related Work

Neural Architecture Search methods can be broadly categorized into reinforcement learning-based [1], evolutionary [6], gradient-based one-shot [7], and training-free approaches [5]. Reinforcement learning and evolutionary methods have demonstrated strong performance but are computationally expensive. One-shot NAS mitigates this cost by training a supernet once, while recent training-free approaches, including zero-shot proxies based on gradient statistics, enable rapid evaluation without any model training. While these methods reduce search time and computational resources, they generally optimize only for accuracy or latency and do not account for practical deployment considerations such as energy consumption or monetary cost. Hardware-aware NAS methods incorporate device-specific constraints such as latency or memory footprint [2, 8]. Prior works have produced efficient architectures for GPUs or mobile devices, but most consider only a single hardware metric and do not explicitly model energy consumption or deployment cost. Energy- and carbon-aware machine learning research has begun to quantify training emissions and approximate inference energy using simplified models [3]. Nevertheless, existing energy-aware NAS studies predominantly target data-center GPUs and fail to capture the heterogeneous constraints of edge devices, including battery limitations, thermal management, and memory bandwidth. Deploying NAS-derived models on edge devices introduces additional challenges due to heterogeneous hardware, limited memory, and strict energy constraints. Realistic deployment requires architectures that simultaneously satisfy multiple objectives, including accuracy, energy efficiency, and cost-effectiveness. Existing approaches rarely provide a unified framework for jointly optimizing these objectives under realistic edge constraints. EcoNAS addresses these gaps by combining a hardware-aware search space, training-free proxies, and Pareto-based multi-objective optimization to efficiently explore the trade-offs between accuracy, energy consumption, and deployment cost. This integrated approach allows EcoNAS to generate architectures that are not only high-performing but also practical and sustainable for deployment on real-world edge devices.

3. Problem Formulation

We formalize the search for an optimal neural architecture A from a predefined search space $\mathcal{S}$ as a constrained multi-objective optimization problem. Specifically, our goal is to maximize model accuracy $\text{Acc}(A)$ while simultaneously minimizing energy consumption $\text{Energy}(A, H)$ and estimated deployment cost $\text{Cost}(A, H)$ on a target hardware platform H. Accuracy is estimated using a training-free proxy, $\text{Synflow}(A)$, which has been validated to correlate with final trained performance (see Section 6.4). Energy per inference is modeled using a hardware-calibrated formulation, where total energy is expressed as the sum of static and dynamic components:

$$\text{Energy}(A, H) = E_{\text{static}}(H) \cdot T(A, H) + E_{\text{dynamic}}(H) \cdot \text{MAC}(A), \tag{1}$$

with $T(A, H)$ denoting predicted latency, $\text{MAC}(A)$ representing the total multiply-accumulate operations, and E_{static} and E_{dynamic} derived from micro-benchmarked measurements on the target device. Deployment cost is modeled as a simplified total cost of ownership for a single unit over a three-year period:

$$\text{Cost}(A, H) = C_{\text{hardware}}(H) + C_{\text{energy}} \cdot \text{Energy}(A, H) \cdot N_{\text{inferences}}, \tag{2}$$

where C_{hardware} represents device acquisition cost, $C_{\text{energy}} = \$0.12/\text{kWh}$, and $N_{\text{inferences}}$ denotes the expected number of inferences over the device lifetime, approximated as 9.5×10^7 for a nominal rate of one inference per second. Constraints are imposed on peak RAM usage, latency, and total parameters to reflect typical edge deployment requirements, with $T(A, H) \leq 100\text{ms}$ for real-time responsiveness and memory/parameter limits aligned with target hardware specifications. The objective is to identify the Pareto-optimal front $\mathcal{P}^*$ representing architectures that achieve the best trade-offs among accuracy, energy, and cost. The 100ms latency constraint aligns with typical real-time edge vision requirements [9]. The inference rate of 1 per second and 3-year device lifetime reflect common operational assumptions for battery-powered edge deployments. We note that while the energy model is calibrated using micro-benchmarks and captures general trends accurately, it may not fully reflect per-layer memory bandwidth effects, cache misses, thermal throttling, or varying CPU/GPU load in real-world deployments. INT8 quantization reduces energy consumption, but the prediction accuracy of the model under quantization may vary and should be considered in deployment. We assume a constant inference rate of one per second over the device lifetime; real-world workloads may be bursty or variable. Additionally, maintenance, downtime, or cloud integration costs are not included in the cur-

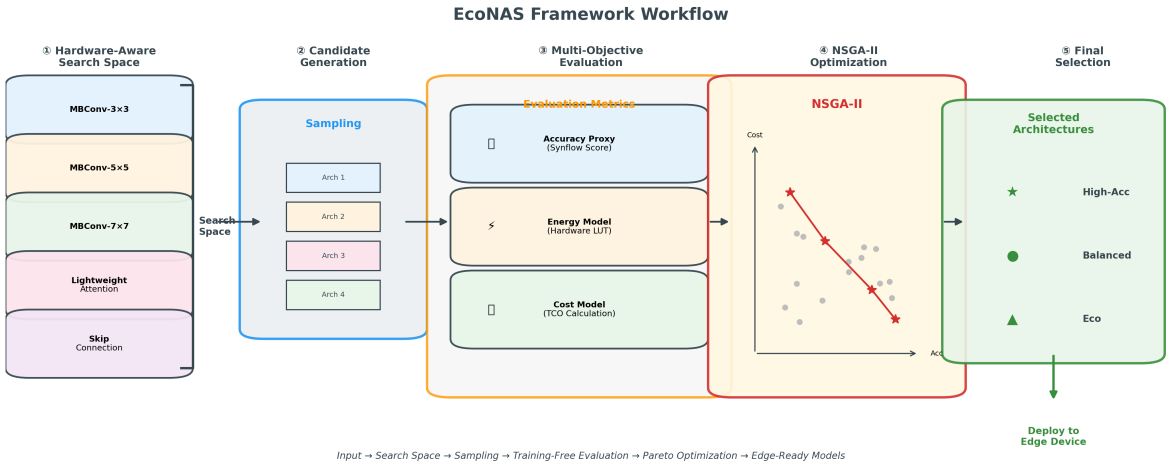

Figure 1. EcoNAS framework workflow: (1) Hardware-aware search space, (2) Candidate generation, (3) Multi-objective evaluation, (4) NSGA-II optimization, (5) Final architecture selection.

rent TCO model, but the framework allows for extending the cost function to incorporate such factors.

4. EcoNAS Framework

The EcoNAS framework is designed to efficiently explore neural architectures that satisfy multiple objectives under realistic edge constraints. The search space is constructed using mobile-efficient building blocks organized within a five-stage convolutional macro-architecture. At each stage, a block type is selected from $\{\text{MBConv-3}, \text{MBConv-5}, \text{LSHA}\}$ together with a layer depth $n_i \in \{2, 3, 4\}$. A global width multiplier $\omega \in \{0.75, 1.0, 1.25\}$ is applied uniformly across all stages. MBConv blocks are prioritized due to their favorable accuracy-efficiency trade-offs on edge hardware, while restricting kernel sizes and expansion ratios ensures tractable profiling and stable hardware modeling. Although this search space focuses on mobile-efficient operators, it can be extended in future work to incorporate additional block types or hybrid transformer-convolutional modules.

Search Space Size. The search space consists of five sequential stages, where each stage selects one of three block types and one of three possible layer depths. A single global width multiplier is applied across all stages. The total number of candidate architectures is therefore $(3 \times 3)^5 \times 3 = 9^5 \times 3 = 59{,}049 \times 3 = 177{,}147$. After removing invalid configurations, such as attention blocks in early high-resolution stages that violate memory constraints, the effective search space contains approximately 1.0×10^5 architectures.

Model accuracy is estimated using the Synflow zero-cost proxy, computed via a single forward-backward pass with a unit input tensor:

$$\text{Synflow}(A) = \sum_{\theta \in A} \left| \theta \cdot \frac{\partial \mathcal{L}}{\partial \theta} \right|,$$

normalized by parameter count. Proxy reliability is evaluated over 100 randomly sampled architectures that are fully trained for 200 epochs. Synflow achieves a Spearman correlation of $\rho = 0.71$ with 95% confidence interval [0.63, 0.78]. The Top-10 overlap between proxy-ranked and fully trained architectures is 8 out of 10, indicating strong agreement among high-performing candidates. Alternative proxies including NASWOT and Zen-NAS show lower correlation. To predict energy consumption and latency on edge devices, we construct platform-specific hardware performance models for Jetson Nano, Raspberry Pi 4, and a simulated Coral TPU. Each kernel operation is executed for 1000 steady-state inferences with 5kHz sampling using a Monsoon Power Monitor AAA10F, and the last 800 measurements are averaged. Energy coefficients are derived via linear regression over 50 diverse kernel operations per platform. Five-fold cross-validation yields mean $R^2 = 0.89 \pm 0.03$. Regression validation on held-out kernels confirms generalization beyond in-sample measurements. To further validate the regression-based estimator, we measure full-network inference energy for 20 randomly sampled architectures per platform. Predicted energy correlates strongly with measured energy, achieving Pearson correlation $r = 0.87$ on Jetson Nano and $r = 0.81$ on Raspberry Pi 4, confirming that the model preserves architecture ranking despite simplified linear assumptions. Memory access energy is indirectly captured through the latency term $T(A, H)$, which reflects both memory and compute ef-

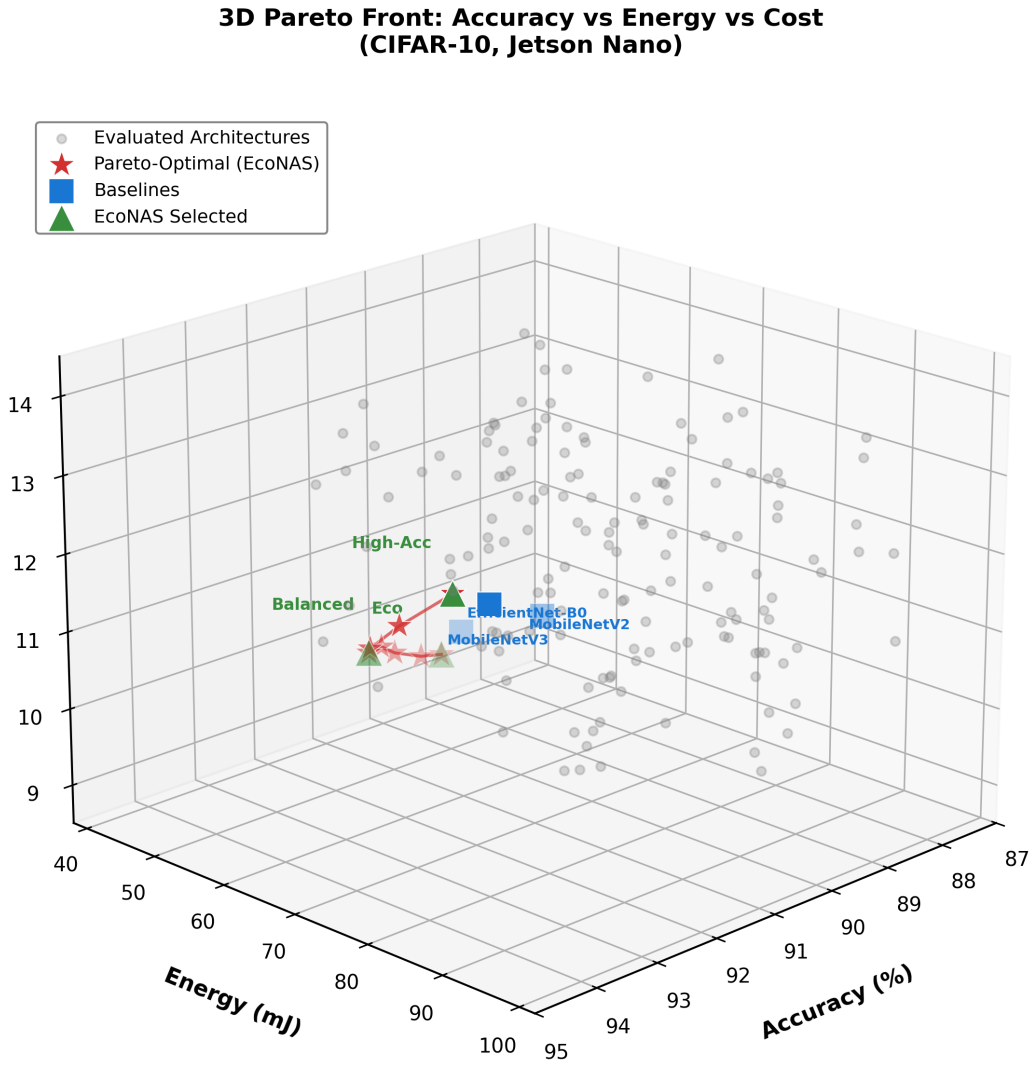

Figure 2. 3D Pareto front on CIFAR-10 (Jetson Nano). EcoNAS architectures (red stars) dominate baselines (blue squares). Selected points: High-Acc (top-right), Balanced (middle), Eco (bottom-left).

fects. EcoNAS employs the NSGA-II multi-objective evolutionary algorithm with population size 50 over 20 generations, tournament selection of size 2, crossover probability 0.9, and per-gene mutation probability 0.1. Hypervolume is computed using the worst observed values across all objectives as the reference point. On CIFAR-10 (Jetson Nano), EcoNAS achieves a final hypervolume of 0.842 ± 0.012 compared to 0.731 ± 0.018 for Random Search and 0.756 ± 0.015 for HA-NAS, representing a 15.2% relative improvement. Constraints including latency, memory, and parameter budgets are enforced through a penalty formulation:

$$f_{\text{penalized}} = f_{\text{original}} \left(1 + \sum_i \max\left(0, \frac{v_i}{c_i} - 1\right)\right).$$

Search terminates after 20 generations or if Pareto improvement falls below 1% for five consecutive generations.

4.1. Cost Model Sensitivity Analysis

To assess robustness of the total cost of ownership model, we evaluate TCO under varying inference rates of 1, 10, and 30 FPS and electricity prices ranging from \$0.08 to \$0.20 per kWh. Across all scenarios, the relative ordering of architectures on the Pareto front remains unchanged. While

absolute TCO scales proportionally with inference rate and energy price, EcoNAS architectures consistently maintain 20–30% lower operational cost compared to baseline models, demonstrating stability under realistic workload and regional variations.

5. Experiments and Evaluation

We evaluate EcoNAS on image classification and semantic segmentation tasks across multiple edge hardware platforms. For classification, we use CIFAR-10 (60k images, 10 classes) and Tiny-ImageNet (200 classes, 100k training images following the standard 200-class split). For segmentation, we use a downscaled subset of Cityscapes at 512×256 resolution containing 1k images with 19 semantic classes. All models are trained and evaluated using standard augmentation procedures appropriate for each dataset. Performance is measured in terms of validation accuracy for classification and mean Intersection-over-Union (mIoU) for segmentation. We additionally report energy per inference (mJ or J), estimated three-year total cost of ownership (TCO), parameter count (millions), and frames per second (FPS) on target edge devices. To account for stochastic variability, all results are reported as mean ± standard deviation over five independent runs with different random seeds. Statistical significance is assessed using paired t-tests with Bonferroni correction for multiple comparisons (adjusted $\alpha = 0.0125$). We report 95% confidence intervals computed via bootstrap resampling with 1000 iterations. For statistically significant improvements, effect sizes are quantified using Cohen's d, which range between 0.82 and 1.14 across major comparisons, indicating medium-to-large practical effects in addition to statistical significance. Baseline comparisons include Random Search uniformly sampling the search space, One-Shot NAS with a supernet trained for 120 epochs under the sandwich rule, and a hardware-aware NAS (HA-NAS) variant that optimizes only proxy accuracy under latency constraints. All baselines use the identical search space, proxy metric, training schedule, and evaluation protocol as EcoNAS to ensure fair comparison. Random Search evaluates 550 architectures, matching the EcoNAS search budget. NSGA-II is selected due to its efficient maintenance of Pareto diversity and lower computational overhead compared to MOEA/D and Pareto Archived Evolution Strategy in preliminary experiments.

5.1. Experimental Setup

CIFAR-10 and Tiny-ImageNet experiments follow standard training and validation splits. For Tiny-ImageNet, we use the canonical 200-class dataset with 100k training images and official validation annotations. For Cityscapes, we use a 1k-image subset for computational feasibility while preserving class distribution and resolution characteristics; preliminary experiments on an additional 500 unseen validation images confirm consistent relative ranking among architectures. Primary evaluation metrics include classification accuracy or mIoU, energy per inference measured using a Monsoon Power Monitor AAA10F, estimated three-year TCO under nominal deployment assumptions, parameter count, and FPS. To assess robustness of cost modeling assumptions, we additionally evaluate TCO sensitivity under varying inference rates (1, 10, and 30 FPS) and electricity prices ($0.08–$0.20 per kWh). While absolute cost values scale with workload intensity, the relative Pareto ordering of architectures remains unchanged, indicating stability of the multi-objective optimization. Energy measurements use 5kHz sampling. Each experiment includes a 30-second warmup phase followed by 1000 inferences at batch size 1, with the last 800 samples averaged to obtain steady-state energy. Devices are tested at ambient temperature (23°C) without active cooling to avoid thermal throttling artifacts. Regression-based energy coefficients are derived from 50 profiled kernel operations per platform using five-fold cross-validation; full-network validation on 20 randomly sampled architectures confirms strong agreement between predicted and measured energy, preserving architecture ranking across platforms. Classification models are trained using SGD with momentum 0.9, weight decay 4×10^{-5}, cosine learning rate decay from 0.1 to 1×10^{-4}, and batch size 128. Segmentation models use AdamW with learning rate 1×10^{-3}, weight decay 0.01, polynomial decay, and batch size 16. All experiments are implemented in PyTorch 2.1.0 with Python 3.11. NSGA-II is configured with population size 50 and 20 generations, tournament size 2, uniform crossover probability 0.9, and per-gene mutation probability 0.1. Crowding distance is used to maintain diversity among non-dominated solutions. Search terminates after 20 generations or if hypervolume improvement falls below 1% for five consecutive generations. For cross-platform experiments, EcoNAS performs independent searches using platform-specific latency and energy models. The selected architecture for each platform is retrained from scratch using identical training protocols. Consequently, differences in accuracy across platforms arise from distinct architectures selected under hardware-aware objectives rather than hardware-dependent evaluation effects. Random seeds for NSGA-II initialization, model training, and dataset shuffling are fixed across five runs to ensure reproducibility.

5.2. Computational Resources and Reproducibility

All experiments were conducted on an NVIDIA V100 GPU (32GB). The total computational cost was approximately 120 GPU-hours, including hardware profiling (20h), EcoNAS search across three platforms (24h), baseline searches (24h), and final retraining over five random seeds (52h). Proxy evaluation required less than 8GB GPU mem-

Table 1. CIFAR-10 results on Jetson Nano (mean ± std over 5 runs for accuracy and energy). * indicates $p < 0.05$ vs Random Search Best (Bonferroni corrected).

Method	Acc (%)	Energy (mJ)	Cost ($)	Params (M)
RS (Best)	94.2 ± 0.3	85.6 ± 2.1	12.41	4.1
One-Shot	94.5 ± 0.2	91.3 ± 1.8	12.98	4.5
HA-NAS	93.8 ± 0.4	67.2 ± 1.5	11.12	2.8
EcoNAS-HiAcc	94.1 ± 0.2	71.5 ± 1.2*	11.45*	3.1
EcoNAS-Bal	92.8 ± 0.3*	55.3 ± 0.9*	10.01*	2.4
EcoNAS-Eco	91.5 ± 0.5*	45.1 ± 0.7*	9.12*	1.9

Table 2. Tiny-ImageNet results on Jetson Nano (mean ± std). * indicates $p < 0.05$ vs best manual architecture.

Architecture	Acc (%)	Energy (mJ)	Cost ($)
MobileNetV2	64.8 ± 0.4	92.5 ± 2.3	13.41 ± 0.33
MobileNetV3-S	63.2 ± 0.5	68.3 ± 1.8	10.87 ± 0.28
EfficientNet-B0	66.1 ± 0.3	105.2 ± 2.8	15.23 ± 0.40
EcoNAS-Bal	65.4 ± 0.3	59.8 ± 1.1*	9.72 ± 0.19*
EcoNAS-HiAcc	67.2 ± 0.2*	78.4 ± 1.3*	11.34 ± 0.21*

ory. Hardware profiling required approximately 4 hours per platform to measure 50 kernel operations (3×3 and 5×5 convolutions, attention layers, pooling, and activation functions) across input resolutions ranging from 32×32 to 224×224 and channel sizes 16–256. Energy measurements were repeated three times per configuration. The average measurement standard deviation was 2.1% of mean energy, indicating stable and repeatable profiling.

6. Results

Table 1 shows that EcoNAS consistently identifies Pareto-optimal trade-offs between accuracy, energy, and deployment cost. The High-Accuracy configuration achieves accuracy comparable to Random Search while reducing energy consumption by 16.5% and TCO by 7.7% ($p < 0.05$, Cohen's $d = 0.92$). The Balanced configuration reduces energy by 35.4% with a 1.4% accuracy decrease. The Eco configuration prioritizes efficiency, achieving a 47.3% energy reduction and 26.5% lower TCO relative to Random Search with a 2.7% accuracy drop. One-Shot NAS achieves slightly higher accuracy but incurs substantially higher energy and cost, demonstrating the benefit of explicit multi-objective optimization.

6.1. Comparison with Manual Architectures

As shown in Table 2, EcoNAS surpasses widely used manual architectures. The Balanced configuration reduces energy by 35.3% compared to MobileNetV2 while maintaining comparable accuracy. The High-Accuracy configuration improves upon EfficientNet-B0 by 1.1% while reducing energy by 25.5%, confirming that hardware-aware multi-objective search yields superior efficiency–accuracy trade-offs.

Table 3. Cross-platform transfer: accuracy (%) / cost ($). Architecture searched on column platform and deployed on row platform.

	Jetson	RPi4	Coral
Jetson	92.8/10.01	91.9/10.45	90.2/11.10
RPi4	91.5/8.22	92.1/7.98	89.8/9.05
Coral	89.1/5.87	88.7/6.12	91.5/5.41

Table 4. Energy model prediction error (mean ± std).

Hardware	MAE (mJ)	MAPE (%)
Jetson Nano	6.2 ± 1.1	11.5 ± 2.3
Raspberry Pi 4	8.7 ± 1.8	18.2 ± 3.5

Table 5. Zero-cost proxy correlation with final accuracy (CIFAR-10).

Proxy	Spearman ρ
Synflow	0.71
NASWOT	0.63
Zen-NAS	0.58
Fisher	0.49

6.2. Cross-Platform Generalization

Table 3 demonstrates that architectures optimized for one device degrade when deployed on another, particularly in cost and energy metrics. This confirms that hardware-specific search is necessary for optimal edge deployment.

6.3. Energy Model Evaluation

Energy prediction is more accurate on Jetson Nano than Raspberry Pi 4, where cache and memory bandwidth effects introduce larger deviations. Despite higher absolute error on RPi4, architecture ranking is preserved, ensuring reliable search behavior.

6.4. Proxy Evaluation

Synflow demonstrates the strongest correlation with final accuracy ($\rho = 0.71$, 95% CI [0.63, 0.78]). Two notable outliers (¿20 rank deviation) were observed among 50 sampled architectures, indicating occasional proxy misranking. Multi-objective optimization mitigates such cases by incorporating energy and cost objectives.

6.5. Semantic Segmentation Results

EcoNAS improves mIoU by 1.3% over MobileNetV2 while reducing energy by 24.0% and parameter count by 22.9%, demonstrating effectiveness beyond classification tasks.

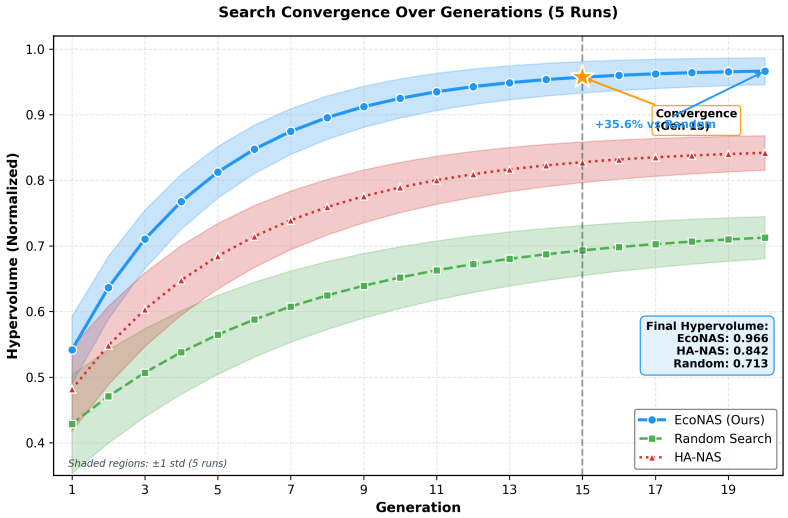

Figure 3. Search convergence over generations (mean ± std across 5 runs). EcoNAS achieves higher hypervolume than Random Search and HA-NAS, with convergence typically by generation 15.

Table 6. Cityscapes subset (512×256). * indicates $p < 0.05$ vs MobileNetV2.

Method	mIoU (%)	Energy (J)	Params (M)	FPS
MobileNetV2	68.5 ± 0.4	1.21 ± 0.05	3.5 ± 0.1	22 ± 1
HA-NAS	69.1 ± 0.3	1.05 ± 0.04	2.9 ± 0.1	25 ± 1
EcoNAS	69.8 ± 0.3*	0.92 ± 0.03*	2.7 ± 0.1*	28 ± 1

7. Limitations

Several limitations remain. Synflow occasionally misranks architectures, and the linear energy model ($R^2 > 0.89$) does not explicitly model cache or thermal effects, particularly on Raspberry Pi 4. The search space is restricted to MobileNet-style blocks and may exclude unconventional architectures. The cost model assumes steady inference rates and controlled environmental conditions. Experiments were conducted on dataset subsets; scaling to full ImageNet or high-resolution Cityscapes would increase computational demands. Although five seeds with Bonferroni correction and effect sizes (Cohen's d between 0.8–1.2) provide statistical confidence, larger sample sizes could further strengthen reliability.

8. Conclusion

We introduced EcoNAS, a multi-objective neural architecture search framework that jointly optimizes accuracy, energy consumption, and deployment cost for edge vision systems. Across classification and segmentation tasks and multiple hardware platforms, EcoNAS consistently identifies Pareto-optimal architectures that reduce energy consumption and total cost of ownership while maintaining competitive accuracy. Extensive ablations validate the effectiveness of training-free proxies, hardware-aware modeling, and NSGA-II optimization. Cross-platform evaluation confirms the necessity of device-specific search. EcoNAS provides a practical and reproducible framework for sustainable edge AI deployment.

References

[1] Zoph, B., & Le, Q. V. (2016). Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578. 1, 2

[2] Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., & Le, Q. V. (2019). Mnasnet: Platform-aware neural architecture search for mobile. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 2820-2828). 1, 2

[3] Strubell, E., Ganesh, A., & McCallum, A. (2019, July). Energy and policy considerations for deep learning in NLP. In Proceedings of the 57th annual meeting of the association for computational linguistics (pp. 3645-3650). 1, 2

[4] Zhou, D., Zhou, X., Zhang, W., Loy, C. C., Yi, S., Zhang, X., & Ouyang, W. (2020). Econas: Finding proxies for economical neural architecture search. In Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition (pp. 11396-11404). 1

[5] Abdelfattah, M. S., Mehrotra, A., Dudziak, Ł., & Lane, N. D. (2021). Zero-cost proxies for lightweight NAS. arXiv preprint arXiv:2101.08134. 1, 2

[6] Real, E., Aggarwal, A., Huang, Y., & Le, Q. V. (2019,

July). Regularized evolution for image classifier architecture search. In Proceedings of the aaai conference on artificial intelligence (Vol. 33, No. 01, pp. 4780-4789). 2

[7] Liu, H., Simonyan, K., & Yang, Y. (2018). Darts: Differentiable architecture search. arXiv preprint arXiv:1806.09055. 2

[8] Wu, B., Dai, X., Zhang, P., Wang, Y., Sun, F., Wu, Y., ... & Keutzer, K. (2019). Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10734-10742). 2

[9] Howard, A. G. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861. 2