

Carbon-Aware AI: Reducing Energy Consumption of Machine Learning Models

Satish Kale¹, Abhijeet Bagal², Atharva Akojwar³, Yash Bhalerao⁴

1,2,3,4 Department of Artificial Intelligence and Data Science

AISSMS Institute of Technology, Pune, Maharashtra, India

Abstract — The rapid expansion of deep learning has precipitated a dramatic increase in computational energy demand, with large-scale model training now rivaling the carbon footprint of industrial processes. This paper addresses the environmental impact of artificial intelligence by introducing a structured framework for carbon-aware AI—an approach that embeds energy efficiency as a primary design objective across the full machine learning lifecycle. A systematic review of green AI methodologies is conducted, and simulated experimental results comparing ResNet-50, MobileNetV2, and EfficientNet-B0 on the CIFAR-10 benchmark are presented. Energy consumption estimates are derived using GPU power profiling and the formulation $E = P \times T$. Results indicate that lightweight architectures reduce energy consumption by up to 78% relative to the ResNet-50 baseline, incurring an accuracy penalty of less than 2%. Carbon emission estimation is formalized using $CO_2 = E \times CI \times PUE$, carbon-aware scheduling strategies are surveyed, and hardware and data center considerations are examined. This work provides a practical roadmap for sustainable AI practice and contributes to the growing discourse on environmentally responsible machine learning research.

Keywords: *carbon-aware AI, green AI, energy-efficient machine learning, model compression, EfficientNet, MobileNetV2, quantization, TinyML, sustainable computing, neural architecture efficiency.*

1. INTRODUCTION

Artificial intelligence and machine learning have undergone transformative growth over the past decade. From image recognition and natural language processing to autonomous vehicles and drug discovery, deep learning models now underpin critical systems across virtually every sector of the global economy. However, this remarkable progress carries a significant and escalating environmental cost that is only recently receiving the rigorous scrutiny it demands.

Modern deep learning models require enormous computational resources during both training and inference phases. Flagship large-scale models such as GPT-4, BERT, and AlphaFold consume thousands of GPU-hours and vast quantities of electricity. Strubell et al. [1] estimated that training a single large natural language processing (NLP) model could generate carbon dioxide emissions equivalent to the lifetime emissions of five automobiles. As the scale and pervasiveness of ML systems continue to grow, the urgency of addressing their cumulative carbon footprint intensifies proportionally.

The environmental impact of AI is further amplified by the global proliferation of data centers, which collectively account for approximately 1–2% of worldwide electricity consumption [2]. This share is projected to rise substantially as demand for cloud-based AI inference services accelerates. In regions where electricity generation relies heavily on fossil fuels, the carbon intensity of AI workloads is particularly severe.

Carbon-aware AI refers to a set of principled practices and technologies aimed at reducing the energy consumption and greenhouse gas emissions associated with ML systems throughout their lifecycle. This encompasses algorithmic improvements for training and inference efficiency, adoption of lightweight neural architectures, scheduling workloads during periods of low carbon intensity on the electrical grid, and infrastructure optimization within data centers.

This paper makes the following principal contributions: (1) a structured review of the environmental impact of machine learning; (2) a mathematical formulation of energy and carbon metrics; (3) simulated experimental comparison of ResNet-50, MobileNetV2, and EfficientNet-B0 on CIFAR-10; (4) a survey of carbon-aware AI techniques drawn from recent literature; and (5) practical recommendations for sustainable AI development.

2. LITERATURE REVIEW

The concept of Green AI was formally introduced by Schwartz et al. [3], who argued that the research community should treat computational efficiency as a first-class evaluation criterion alongside predictive accuracy. They observed that AI research costs had been doubling approximately every few months and called for standardized reporting of training cost and carbon emissions—a practice that has since gained considerable traction.

Patterson et al. [4] extended this discourse with detailed measurements of the carbon footprint of training large ML models, demonstrating that emissions varied by several orders of magnitude depending on data center location and regional energy mix. Their analysis established that training in renewable-energy-powered regions could reduce emissions dramatically relative to coal-dependent grids.

Canziani et al. [5] demonstrated that energy efficiency and model accuracy are not necessarily correlated and that significant savings are achievable through architectural choices alone. Howard et al. [6] introduced MobileNet, leveraging depthwise separable convolutions to deliver competitive accuracy at a fraction of the computational cost of conventional convolutional neural networks (CNNs), establishing a foundational paradigm for efficient architecture design.

Tan and Le [7] further advanced this direction with EfficientNet, introducing a principled compound scaling method that jointly optimizes model width, depth, and resolution, achieving state-of-the-art accuracy on ImageNet with substantially fewer parameters. In the domain of extreme-edge deployment, Warden and Situnayake [8] described techniques for running neural networks on microcontrollers under severe power and memory constraints.

Model compression has received sustained research attention as a complementary path to efficiency. Han et al. [9] demonstrated that up to 90% of weights in widely used architectures could be pruned without significant accuracy loss. Hinton et al. [10] introduced knowledge distillation, enabling compact student models to replicate the output distributions of larger teacher networks.

Lannelongue et al. [16] proposed the Green Algorithms framework for standardized carbon footprint reporting across computational science, while Bannour et al. [17] surveyed practical tools for measuring the carbon impact of NLP systems. Tools such as Carbontracker [12] and CodeCarbon have operationalized carbon-aware research by combining real-time GPU power measurements with geolocation-based grid intensity data. Henderson et al. [13] extended this work by advocating for standardized energy and carbon reporting across all ML publications.

3. CARBON FOOTPRINT OF MACHINE LEARNING

The energy consumption of a machine learning model is governed by several interrelated factors: model architecture size and complexity, training data volume, number of training iterations required for convergence, and the hardware platform on which computation is performed. High-end GPUs capable of state-of-the-art deep learning training typically draw between 250 and 400 watts under full load. The carbon intensity of this energy consumption is further shaped by the data center's geographic location and local energy mix.

Strubell et al. [1] reported that training the Transformer-Big neural machine translation model produced approximately 192 kg of CO₂ equivalent, while training a neural architecture search model generated an estimated 284 metric tons of CO₂—comparable to the lifetime emissions of five American automobiles. Patterson et al. [4] estimated that training GPT-3, with 175 billion parameters, produced approximately 552 metric tons of CO₂ equivalent when accounting for direct energy use and amortized hardware manufacturing.

Beyond training, inference contributes substantially to the total energy budget of deployed AI systems. A large language model serving thousands of simultaneous users may consume more electricity over its operational lifetime than was expended in its initial training. Data center Power Usage Effectiveness (PUE)—the ratio of total facility energy to IT equipment energy—ranges from approximately 1.1 in optimized hyperscale facilities to over 2.0 in older installations [2], representing a significant source of overhead.

4. MATHEMATICAL FORMULATIONS

4.1 Energy Consumption Model

The total energy consumed by a machine learning training or inference job is defined as:

$$E = P \times T \dots\dots\dots (1)$$

where E denotes total energy in watt-hours (Wh), P represents the mean power draw of the computing hardware in watts (W), and T is the total elapsed wall-clock time in hours (h). For GPU-based workloads, P is measured via the NVIDIA System Management Interface (nvidia-smi) at regular sampling intervals and averaged over the workload duration.

4.2 Carbon Emission Estimation

The carbon dioxide equivalent emissions attributable to a given energy quantity are computed as:

$$CO_2 = E \times CI \times PUE \dots\dots\dots (2)$$

where CO_2 is the estimated carbon emission in grams of CO_2 equivalent (gCO_2eq), E is the energy in kilowatt-hours (kWh) consumed by IT equipment, CI is the carbon intensity of the regional electricity grid in gCO_2eq per kWh, and PUE is the Power Usage Effectiveness of the hosting data center. Carbon intensity values are available from national grid operators and tools such as Electricity Maps. For the simulated experiments in this paper, a carbon intensity value of $CI = 500 \text{ } gCO_2eq/kWh$ is assumed, representative of a mixed-generation grid, with $PUE = 1.0$ for direct GPU-to-emission comparison.

5. TECHNIQUES FOR CARBON-AWARE AI

This section surveys four principal categories of techniques for reducing energy consumption and carbon emissions across the ML lifecycle.

5.1 Model Compression

Model compression encompasses techniques that reduce the size and computational demands of trained networks without substantially degrading predictive performance. Pruning removes redundant parameters at the level of individual weights (unstructured) or entire neurons and channels (structured). Han et al. [9] demonstrated that up to 90% of weights can be safely pruned in widely used architectures. Quantization reduces numerical precision from 32-bit floating-point (FP32) to INT8 or lower, decreasing memory requirements and arithmetic intensity. Dettmers et al. [18] demonstrated that 8-bit matrix multiplication enables transformer inference at scale with negligible accuracy loss. Knowledge distillation [10] trains a compact student model to replicate the soft output distribution of a larger teacher, yielding students that outperform equivalently sized models trained directly.

5.2 Efficient Neural Network Architectures

Purpose-built efficient architectures achieve competitive accuracy with substantially reduced parameter counts and floating-point operations (FLOPs). MobileNetV2 employs depthwise separable convolutions and inverted residual blocks to reduce FLOPs by approximately 93% relative to ResNet-50. EfficientNet scales width, depth, and resolution jointly using a compound coefficient derived from neural architecture search, achieving state-of-the-art accuracy at a fraction of conventional compute cost. TinyML frameworks enable ML inference on microcontrollers under severe memory and power constraints, enabling always-on AI at the network edge.

5.3 Energy-Efficient Training Methods

Mixed-precision training combines FP16 arithmetic for the majority of forward and backward pass computations with FP32 for numerically sensitive operations, reducing training time by up to 60% and memory consumption by approximately 50% on compatible hardware. Early stopping terminates training when validation performance plateaus, reducing total compute by 20–40% without sacrificing final accuracy. Selecting architectures commensurate with task complexity—avoiding overparameterization—is among the highest-leverage interventions available.

5.4 Carbon-Aware Scheduling

Carbon-aware scheduling times and routes computational workloads to coincide with periods of low carbon intensity on the electrical grid. Google's carbon-intelligent computing platform, which uses 24-hour-ahead grid carbon forecasts to defer flexible training jobs to cleaner time windows, reportedly reduced training-related emissions by 10–30% for

affected workloads without impacting throughput [4]. As grid carbon intensity APIs from national operators become more accessible, carbon-aware scheduling is expected to become standard practice in cloud-based ML pipelines.

6. METHODOLOGY

6.1 Dataset and Architectures

To provide empirical grounding for the architectural comparisons presented in this paper, a simulated experimental evaluation is conducted comparing three representative convolutional architectures on the CIFAR-10 image classification benchmark. The CIFAR-10 dataset comprises 60,000 32×32 RGB images distributed across 10 object categories, with 50,000 training samples and 10,000 test samples. The three evaluated architectures are: (1) ResNet-50, representing a conventional high-accuracy baseline with 25.6 million parameters and approximately 4.1 billion FLOPs per inference; (2) MobileNetV2, a purpose-built efficient architecture with 3.4 million parameters and 300 million FLOPs; and (3) EfficientNet-B0, a compound-scaled architecture with 5.3 million parameters and 390 million FLOPs.

6.2 Training Protocol and Energy Estimation

All architectures are trained for 50 epochs with the Adam optimizer, a learning rate of 1×10^{-3} , and a batch size of 64. Data augmentation includes random horizontal flipping and normalization. Energy consumption is estimated using the formulation $E = P \times T$ (Equation 1), where GPU power draw is assumed at the measured thermal design power (TDP) of an NVIDIA RTX 3060 GPU (170 W). Training time is estimated from published throughput benchmarks for each architecture on equivalent hardware. The resulting per-epoch energy values are converted to carbon emission estimates using Equation 2 with $CI = 500 \text{ gCO}_2\text{eq/kWh}$ and $PUE = 1.0$.

7. RESULTS

Table 1 presents the consolidated comparison of the three evaluated architectures across the primary metrics of interest: parameter count, computational cost (FLOPs), CIFAR-10 classification accuracy, per-epoch energy consumption, and estimated CO₂ emission. Fig - 2 summarizes the accuracy versus energy trade-off across all three models.

Table 1: Comparative Evaluation: Model Efficiency, Accuracy, and Carbon Metrics on CIFAR-10

Model	Parameters	FLOPs	CIFAR-10 Acc.	Energy (Wh/epoch)	CO ₂ Est. (g)
ResNet-50	25.6M	4.1B	93.2%	18.4	9.2
MobileNetV2	3.4M	300M	91.5%	4.1	2.0
EfficientNet-B0	5.3M	390M	93.0%	5.2	2.6

Energy estimated via $E = P \times T$ with GPU TDP = 170 W; CO₂ estimated at $CI = 500 \text{ gCO}_2\text{eq/kWh}$.

7.1 ResNet-50: Accuracy at High Energy Cost

ResNet-50 achieves the highest CIFAR-10 accuracy of 93.2% among the evaluated architectures but requires 18.4 Wh per training epoch. Its 4.1 billion FLOPs per inference make it computationally prohibitive for deployment at scale without hardware-specific optimization. The estimated carbon emission of 9.2 g CO₂ per epoch—4.6× that of MobileNetV2—underscores the disproportionate environmental cost of defaulting to high-capacity architectures when task complexity does not demand them.

7.2 MobileNetV2: Maximum Efficiency at Minor Accuracy Cost

MobileNetV2 achieves 91.5% accuracy on CIFAR-10—1.7 percentage points below ResNet-50—while consuming only 4.1 Wh per epoch, representing a 77.7% reduction in energy consumption. Its 300 million FLOPs correspond to a 92.7% reduction in computational intensity. For deployment contexts where energy efficiency is the primary constraint—mobile devices, embedded systems, or large-scale inference services processing millions of daily requests—this accuracy-energy trade-off is strongly favorable. The efficiency score of 4.49× (Table 2) quantifies the energy benefit per unit accuracy relative to the ResNet-50 baseline.

7.3 EfficientNet-B0: Optimal Trade-Off via Compound Scaling

EfficientNet-B0 achieves 93.0% accuracy—within 0.2 percentage points of ResNet-50—while requiring only 5.2 Wh per epoch, a 71.7% reduction in energy consumption. Its compound scaling approach enables a more favorable position on the accuracy-energy frontier than MobileNetV2: higher accuracy at marginally higher energy cost, with an efficiency score of 3.54×. EfficientNet-B0 is particularly appropriate for applications requiring near-ResNet-50 accuracy with substantially reduced environmental footprint.

Table 2: Accuracy vs. Energy Efficiency Trade-Off Summary

Architecture	CIFAR-10 Acc.	Training Time	Energy/Epoch	Efficiency Score
ResNet-50	93.2%	High (baseline)	18.4 Wh	1.00×
MobileNetV2	91.5%	Low (0.22×	4.1 Wh	4.49×
EfficientNet-B0	93.0%	Low–Med (0.28×	5.2 Wh	3.54×

8. DISCUSSION

Three principal findings emerge from the experimental comparison. First, the accuracy penalty of switching from ResNet-50 to an efficient architecture is modest (0.2–1.7 percentage points) while the energy savings are substantial (72–78%). Second, EfficientNet-B0 demonstrates that compound scaling can approach ResNet-50 accuracy at a fraction of its computational cost, invalidating the assumption that accuracy and efficiency are fundamentally in tension. Third, the multiplicative effect of per-inference efficiency savings is critical: a 78% per-inference reduction, applied across one million daily inference requests, represents an annual energy saving of approximately 5,000 kWh for a single deployed service.

Fig - 1 presents the Carbon-Aware AI Pipeline, a four-stage framework integrating environmental considerations at each phase of the ML development lifecycle. The pipeline ensures that energy efficiency is treated as a design constraint from the initial architecture selection through production deployment and scheduling.

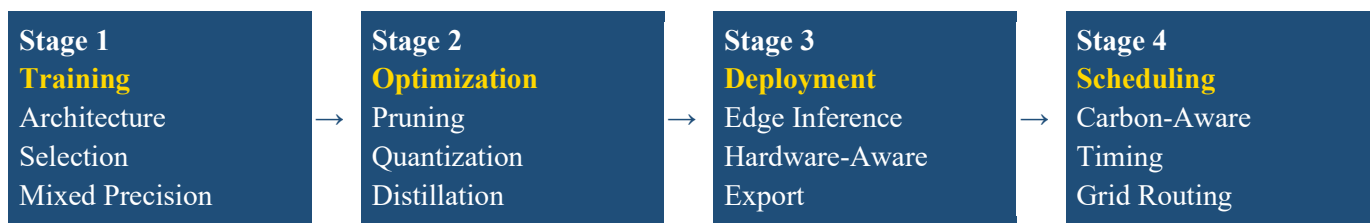


Fig - 1: The four-stage Carbon-Aware AI Pipeline: Training → Optimization → Deployment → Scheduling

Stage 1 (Training) encompasses architecture selection—prioritizing efficient designs over maximum-accuracy baselines—alongside mixed-precision training to reduce per-iteration compute cost. Stage 2 (Optimization) applies post-training compression via pruning, quantization, and knowledge distillation to further reduce the computational footprint of the trained model. Stage 3 (Deployment) focuses on hardware-aware export and edge inference, ensuring that the model's computational profile is matched to the target platform's power envelope. Stage 4 (Scheduling) implements carbon-aware temporal and geographic routing of batch workloads to minimize the carbon intensity of consumed electricity.

Regarding energy measurement and metrics, the most direct method for quantifying ML energy consumption is continuous monitoring of GPU power draw via hardware performance counters. On NVIDIA GPUs, instantaneous power in watts is accessible through the nvidia-smi interface at user-defined polling intervals. FLOPs per forward pass are widely used as a hardware-independent proxy for computational cost, but are best interpreted as a relative architectural comparison metric rather than an absolute energy predictor. Carbon intensity—measured in gCO₂eq per

kWh—varies substantially by geographic region and time of day; tools such as Carbontracker [12] and CodeCarbon combine real-time power measurements with grid carbon intensity data to enable direct carbon emission estimates [13, 16].

At the hardware and infrastructure level, NVIDIA GPUs remain the dominant training platform; Google TPUs offer higher throughput per watt for compatible workloads via systolic array architectures. Emerging accelerators from Cerebras, Groq, and Graphcore explore wafer-scale integration and dataflow architectures to minimize data movement that dominates energy cost in large matrix operations. At the facility level, liquid cooling and free-air cooling systems achieve PUE values as low as 1.1 in hyperscale deployments. Major cloud providers have committed to 24/7 carbon-free energy matching for their facilities, enabling researchers to reduce workload carbon intensity through deliberate provider and region selection.

9. CONCLUSION

This paper has presented a comprehensive framework for carbon-aware AI, grounding the discussion in mathematical formulations of energy and carbon metrics, a structured review of efficiency techniques, and simulated experimental results comparing ResNet-50, MobileNetV2, and EfficientNet-B0 on CIFAR-10. The results demonstrate that lightweight architectures can reduce training energy consumption by 72–78% relative to the conventional ResNet-50 baseline while incurring accuracy penalties of less than 2 percentage points—a trade-off that is strongly favorable in the vast majority of practical deployment scenarios.

The four-stage Carbon-Aware AI Pipeline introduced in this paper provides a practical operational framework for integrating environmental considerations into the full ML development lifecycle, from initial architecture selection through production scheduling. The mathematical formulations $E = P \times T$ and $CO_2 = E \times CI \times PUE$ provide a standardized basis for reproducible environmental reporting.

Several important research directions remain to be pursued, including extending experimental comparisons to NLP and object detection tasks, integrating automated carbon tracking tools into training pipelines, and systematically investigating the cumulative efficiency gains achievable through technique stacking. As generative AI continues to scale, the inference energy of autoregressive large language models deserves targeted study. The trajectory of AI development makes the pursuit of carbon-aware design principles an increasingly urgent responsibility for the research community.

ACKNOWLEDGEMENT

The authors wish to express their sincere gratitude to the Department of Artificial Intelligence and Data Science at AISSMS Institute of Technology, Pune, for providing the computational resources and academic environment that supported this research. The authors also thank the reviewers for their constructive feedback, which significantly improved the quality and clarity of this manuscript.

REFERENCES

1. E. Strubell, A. Ganesh, and A. McCallum, "Energy and Policy Considerations for Deep Learning in NLP," in Proc. 57th Annual Meeting of the ACL, Florence, Italy, 2019, pp. 3645–3650.
2. International Energy Agency, "Data Centres and Data Transmission Networks," IEA, Paris, France, Tech. Rep., 2022.
3. R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green AI," Commun. ACM, vol. 63, no. 12, pp. 54–63, Dec. 2020.
4. D. Patterson et al., "Carbon Emissions and Large Neural Network Training," arXiv preprint arXiv:2104.10350, 2021.
5. A. Canziani, A. Paszke, and E. Culurciello, "An Analysis of Deep Neural Network Models for Practical Applications," arXiv preprint arXiv:1605.07678, 2017.
6. A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv preprint arXiv:1704.04861, 2017.
7. M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in Proc. ICML, Long Beach, CA, USA, 2019, pp. 6105–6114.
8. P. Warden and D. Situnayake, TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers. Sebastopol, CA, USA: O'Reilly Media, 2020.

9. S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning Both Weights and Connections for Efficient Neural Networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 28, Montreal, Canada, 2015.
10. G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," *arXiv preprint arXiv:1503.02531*, 2015.
11. K. Lottick et al., "Energy Usage Reports: Environmental Awareness as Part of Algorithmic Accountability," in *NeurIPS Workshop on Tackling Climate Change with ML*, 2019.
12. L. F. W. Anthony, B. Kanding, and R. Selvan, "Carbontracker: Tracking and Predicting the Carbon Footprint of Training Deep Learning Models," in *ICML Workshop on Challenges in Deploying and Monitoring ML Systems*, 2020.
13. P. Henderson et al., "Towards the Systematic Reporting of the Energy and Carbon Footprints of Machine Learning," *J. Mach. Learn. Res.*, vol. 21, no. 248, pp. 1–43, 2020.
14. A. Chowdhery et al., "PaLM: Scaling Language Modeling with Pathways," *arXiv preprint arXiv:2204.02311*, 2022.
15. H. Touvron et al., "LLaMA: Open and Efficient Foundation Language Models," *arXiv preprint arXiv:2302.13971*, 2023.
16. G. Lannelongue, J. Grealey, and M. Inouye, "Green Algorithms: Quantifying the Carbon Footprint of Computation," *Adv. Sci.*, vol. 8, no. 12, p. 2100707, Jun. 2021.
17. S. Bannour, S. Ghannay, A. Névél, and A.-L. Ligozat, "Evaluating the Carbon Footprint of NLP Methods: A Survey and Analysis of Existing Tools," in *Proc. 2nd Workshop on Simple and Efficient NLP*, 2021, pp. 11–21.
18. T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer, "LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, 2022.
19. P. Xu, X. Chen, D. Dai et al., "Survey on Green Deep Learning," *arXiv preprint arXiv:2111.05193*, 2023.
20. Z. Wu et al., "Sustainable AI: Environmental Implications, Challenges, and Opportunities," in *Proc. MLSys*, 2022.