Article

Towards Green AI: A Lightweight Experimental Study on Carbon-Aware Neural Network Training

Basab Nath [1,*], Fatima Ahmed Mohamed [2], and Farrukh Hassan [3,4]

[1] School of Computer Science and Engineering, Bennett University, Greater Noida, Uttar Pradesh 201310, India
[2] Department of Information and Communication Technology, Faculty of Computing and Information Technology, Tunku Abdul Rahman University of Management and Technology, Kuala Lumpur 53300, Malaysia
[3] School of Computing and Artificial Intelligence, Faculty of Engineering and Technology, Sunway University, No. 5, Jalan Universiti, Bandar Sunway, Selangor Darul Ehsan 47500, Malaysia
[4] Sunway Centre for Electrochemical Energy and Sustainable Technology (SCEEST), School of Engineering & Technology, Sunway University, No. 5, Jalan Universiti, Bandar Sunway, Petaling Jaya, Selangor Darul Ehsan 47500, Malaysia

* Correspondence: basabnath@gmail.com (B.N.)

Abstract

Industry 5.0 places emphasis on human-centered innovation, aligning digital transformation with sustainability, resilience, and societal well-being. Smart cities serve as practical environments where advanced technologies must balance operational efficiency with environmental responsibility. Since October 2023, deep neural networks have been widely used in urban decision-making, including traffic optimization and environmental monitoring. However, the high energy demand required to train large-scale models has raised concerns regarding both environmental impact and economic cost, leading to growing interest in sustainable AI practices. This study explores Carbon-Aware Neural Network Optimization within the Green AI framework as a practical approach to reduce computational carbon emissions while maintaining predictive performance. Three complementary strategies are integrated into a unified training pipeline and monitored using the CodeCarbon framework: partial layer freezing, quantization-aware training, and adaptive early stopping. Experiments are conducted on CIFAR-10 using MobileNetV2 and on a subset of ImageNet with ResNet-50. Results show a reduction in CO_2 emissions of 52% and 49%, respectively, with only minor changes in model accuracy. The findings indicate that combining these techniques provides an effective balance between efficiency and predictive reliability, supporting the development of sustainable AI solutions for Industry 5.0 and smart city applications.

Keywords: Industry 5.0; Green AI; Energy-aware training; Neural networks; CodeCarbon; Sustainability; Quantization-aware training; Early stopping; Runtime layer freezing

Citation: Nath, B.; Mohamed, F.A.; Hassan, F. Towards Green AI: A Lightweight Experimental Study on Carbon-Aware Neural Network Training. *Impact in Computics* 2026, **2**, 1. https://doi.org/10.65500/computics-2026-001

Received: 11 November 2025 | Revised: 28 December 2025 | Accepted: 7 January 2026 | Published: 23 January 2026

1. Introduction

It is obvious that the Industry 5.0 represents a new trend in development of technologies and organizations accents on the digital advancement being based on the sustainable, resilient and human experience idea The new step from which we need to understand innovation– and relinquish the more automatic or data led view that this term might otherwise evoke; is a collaborative act between human beings and autonomous machines, where productivity gains hold account of planetary boundaries. The best possible view of this balance was only evident with the sharp rise of smart cities, where digital infrastructure should not just be functional but also accountable to enhance human and planetary wellbeing [1]. AI (particularly Deep Neural Networks (DNNs)) is the focus in this process, and data-driven decisions based on the AI can be found everywhere today, from Manufacturing to healthcare, logistics, or Energy management. But the explosive growth of AI creates a paradox: The systems that make cities smarter and greener also devour massive amounts of computer resources. The power consumption to train and run the current DNNs has thus put into question the issue of cost, resource expenditure, and the ecological impact in the long term [2]. Research indicates that massive model training can consume large amounts of electricity, leaving a substantial carbon footprint. Indicatively, Strubell et al. have indicated that one large NLP model would emit as much as five cars in their lifetime [3]. Similarly, the consumption of total CPU-based AI servers grew to more than 40 TWh as compared to less than 2 TWh in 2017 [4]. These numbers highlight the importance of being carbon-conscious when developing and implementing AI systems.

In order to solve this issue, the current work places the concept of Carbon-Aware Neural Network Optimization into the greater context of Industry 5.0 and sustainable urban ecosystem. The objective is to develop and implement machine-learning methods that are environmentally less expensive, do not compromise model precision, or usefulness [5]. This discussion is based on three complementary strategies: runtime layer freezing, quantization-aware training, and adaptive early stopping. Both of them focus on different phases of the learning cycle to reduce unnecessary computation without impairing predictive accuracy. Furthermore, real-time emissions were monitored with the help of the CodeCarbon framework [6], providing the researchers and practitioners with clear-cut information on the efficiency versus accuracy trade-off.

The main contributions of this work are:

- A carbon-aware training pipeline that combines a workflow with runtime layer freezing, quantization-aware training, and adaptive early stopping. This is important because our study focuses on using existing techniques in coordination rather than proposing an optimization algorithm.
- An empirical evaluation on CIFAR-10, an ImageNet subset, and ImageNet-100 using MobileNetV2 and ResNets (both fine-tuning/transfer learning/unlabelled-data scenarios) that assess generalisation across datasets and model capacities.
- We conduct comparative experiments, including ablation studies and baseline integrations, to report the results for each component, and the combined application produces more energy savings than any individual technique.

Statistical analysis on all random seeds shows that reductions in energy consumption and CO_2 emissions are consistent, while predictive performance is statistically equivalent to baseline models.

Besides the technical viewpoint, this paper additionally explores long-term factors on organizational strategy and coverage through carbon aware AI. For example, in the case of sustainability that is part of algorithmic design, institutions/city authorities can derive reduced infrastructure costs from this, enhanced compliance with environmental regulations, as well as greater public trust and increased preparation for future sustainability requirements [7]. At the end of the day, Carbon-Aware AI is more than just optimization methods; it represents a wider ethical and future-oriented framework for constructing smart systems that are efficient but also eco-friendly in nature, as envisioned by Industry 5.0.

2. Related work

The recent proliferation of interest in 'Green 7 AI' may indicate a growing awareness that deep learning has a carbon footprint. While artificial intelligence continues to be an integral part of the industry 5.0 vision, it is important to mention also that the use of computation-demanding artificial intelligence restricts its sustainability objectives (including environmental goals). This vast difference has resulted in most of today's search efforts for AI to focus on methods that reduce carbon footprint from training models as much as possible without losing functionality at the level of prediction. In recent years, 'Green AI' has evolved from a niche interest to the more general idea in the field that algorithm development should focus not only on performance but also on responsible environmental stewardship [10].

Such is the case with 'runtime layer freezing'. Imagine that in a neural network, you had one layer that has already converged: there is no reason to keep updating it, but resources get wasted with the repeated computation of 'gradient updates'. A large collection of training runs on the ImageNet dataset demonstrated that this method could reduce training from a few days to several hours! This idea has been developed in federated learning systems, where the client is allowed to freeze layer by layer. This reduces the amount of local computation and also lessens the communication overhead to the entire network.

A second fairly aggressive technique is quantization-aware training (QAT). QAT (Quantization-Aware Training), on the other hand, minimizes precision during learning (for example, using 8 or even 4-bit integers) rather than compressing weights post-training. This makes your models more robust after they are deployed at low precision. And the less doing to do, the less energy used up, and the faster it gets done; a smaller memory request. As explained in [14], quantization serves to unify the efficiency of software operations with that of hardware.

Adaptive early stopping is also a popular regularization attribute, but sustainably it does more. We stop our training when the validation curve is not improving anymore because running for more epochs after this point has basically no effect and creates unnecessary energy waste. This is particularly impactful in AutoML configurations, where analytics servers often terminate weak runs and thus improve the computation [8].

Another strategy is an integrated framework that unifies several of those ideas, where it consolidates pruning along with quantization and compact network design [9, 14]. One of the techniques is MCUNET, which merges a small arch like MobileNetV2 with targeted compression to make ImageNet-scale tasks run on tiny micro-controllers [10,13]. These hybrid systems mix with Industry 5.0 principles of sustainability for the simple fact that coordination in design outperforms isolated tweaks.

Overviews have tried to tie these threads together in a more comprehensive roadmap to sustainability. Sze et al [14] show how algorithmic optimizations lead to hardware constraints and optimized hyperparameters across multiple applications. Pruning strategies and evaluation metrics are discussed in [9]. The environmental impact of AI methods [5] shows that GreenAI is not a tool but an ecosystem of related approaches from model design to deployment. Research has been conducted to measure the carbon footprint of deep learning training [18], LLM [19], and tools for tracking carbon emissions from ML workloads[20], [21]. However, even with this progress, there remain many gaps. A bulk of studies still evaluate each technique separately or through non-adaptive, unchanging pipelines, using methods defined before the run. But there is very little work in adaptive systems switching among actions like freezing, quantization, and early stopping using live sustainability metrics. Concretely, tools like CodeCarbon [6] already have some ability to track emissions in real time, but very few are being integrated into automated learning loops. Constructing feedback models based on this data and training the model to learn has great promise, as it can evolve Green AI from a largely manual research task into something more self-corrective [23,26]. Perhaps this kind of development is precisely what Industry 5.0 needs in order to chime into socially aware and smart transitioning urban systems for real [24],[25].

3. Methodology

The methodology followed in this study involves the neural networks optimization with carbon sensitivity and real-time monitoring of carbon footprint, that demonstrate how artificial intelligence can be trained based on the sustainability objectives of the Industry 5.0 smart cities. The design of the proposed framework was informed by the conceptual model illustrated in Figure 1. Three optimization methods, which are runtime layer freezing, quantization-aware training, and adaptive early stopping, were trained simultaneously within the CodeCarbon framework. Combined together, they create one energy-efficient training pipeline, reducing the energy consumption of computational power and CO_2 emissions, and maintaining the accuracy of models at a relatively low level needed to apply them to real-world cities.

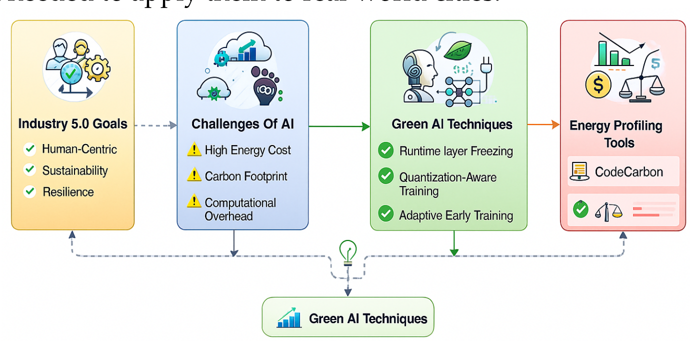

Figure 1. Conceptual framework situating Green AI within the goals and challenges of Industry 5.0.

3.1 Carbon-Conscious Optimization Policies

To minimize the environmental cost of the neural network training without compromising predictive quality, a combination of the corresponding strategies was implemented [27]. The former, runtime layer freezing, is based on the finding that shallow convolutional layers of deep networks tend to converge much sooner than deep layers. When all those initial layers had stabilized (in terms of the validation loss not changing significantly), they were frozen to prevent further gradient updates. This made certain that the GPU did not waste any cycles on layers that could no longer have been used for learning and helped in reducing unnecessary energy usage while preserving the features of the model learnt by it [7, 8].

If the validation loss decreases less than ε for k epochs, freezing of that layer is activated:

$$|L\_val(t) - L\_val(t-1)| < \varepsilon \quad (1)$$

where ε = 0.001 and k = 3. For both Network architectures, the same threshold was used to determine whether convergence had been reached or not (i.e.⇒38%), to allow for an objective comparison between the two network types.

We introduce a new carbon-aware objective function, which permits us to quantitatively model the trade-off between predictive power and environmental cost:

$$J = \text{Accuracy} - \lambda \times \text{Carbon} \quad (2)$$

where Accuracy is the validation accuracy, Carbon is the CO_2 emissions (in grams), and λ is a weighting factor that can be tuned to penalize the carbon emission. In this work, we chose $\lambda = 0.01$ to retain the performance while achieving energy efficiency of PERTa. Training is halted when additional epochs do not lead to improvement of the objective J.

The model was originally set empirically based on prototype experiments for finding runtime layer freezing convergence criteria across both architectures. For example, Layer freezing is initiated when the absolute change in validation loss in three successive epochs is less than 0.001, which depicts that features are converging and stabilized. This threshold is an optimistically large, conservative enough to never have an early freeze, while not wasting on redundant gradient updates. The same convergence criterion (ε = 0.001, k = 3 in all cases) was used for the model trained over lightweight (MobileNetV2) and deep architectures (Resnet-50) to ensure fair comparison and reproducible results [15].

The second approach that was used in this work to reduce arithmetic precision during training is quantization-aware training (QAT). QAT approximates low-precision operations during training, and post-training quantization reduces weights at convergence. We first set off with wider precision setting and set all the l heavy layers to 32-bit floating point. For others, we set it to 8 or 4 for earlier setups. It saves memory and computing time, providing measurable energy savings/carbon emissions with little to no loss of accuracy [3, 14].

Precision assignment followed a layer-wise policy:

Early layers → FP32

Mid layers → INT8

Final layers → FP16

Formally, precision for layer III was defined as:

$$\text{precision}(l) = \text{FP32 if } l < 0.3L \quad (3)$$
$$\text{INT8 if } 0.3L \le l < 0.8L$$
$$\text{FP16 if } l \ge 0.8L$$

where L denotes the total number of layers. This hierarchical precision allocation preserves feature extraction stability while reducing computational cost.

Adaptive early stopping was introduced as the third optimization strategy by incorporating carbon efficiency into the stopping criterion. At each epoch ttt, the change in validation accuracy and energy consumption was computed as:

$\Delta Acc(t) = Acc(t) - Acc(t-1)$

$\Delta Energy(t) = Energy(t) - Energy(t-1)$

Training was terminated when additional computation no longer improved performance but increased energy consumption. The stopping condition was defined as:

$$\Delta Acc(t) < \delta \text{ AND } \Delta Energy(t) > \gamma \quad (4)$$

where δ = 0.1% is the actual threshold of accuracy and γ = 2% presents an increase of energy consumption in relative rate. Training was halted when two consecutive epochs satisfied this condition. This adaptive rule allows better predictive performance while taking into consideration carbon cost by stopping the training when improvement becomes marginal to such an extent that it is done at energy cost [8]. Adaptive means that decisions related to training are made dynamically based on the runtime variations in validation accuracy and energy consumption rather than being determined by a fixed epoch schedule.

3.2 Hyperparameter Selection and Sensitivity

Hyperparameters applied in the proposed pipeline were selected after a preliminary sensitivity analysis with the objective of balancing energy efficiency plus predictive performance. Eq — The trade-off parameter λ in $\lambda \in [0.005, 0.02]$ was tested, where $\lambda = 0.01$ achieved a reasonable balance between retaining accuracy and reducing carbon emissions from (2). For runtime layer freezing, ε ∈ {0.0005, 0.001, 0.002} and k ∈ {2, 3, 4} were explored, where ε = 0.001 and k = 3 ensured stable convergence without premature freezing. For adaptive early stopping, $\delta \in \{0.05\%, 0.1\%, 0.2\%\}$ and $\gamma \in \{1\%, 2\%, 3\%\}$ were tested, with $\delta = 0.1\%$ and $\gamma = 2\%$ yielding consistent energy savings while maintaining comparable accuracy. These values were fixed across experiments to ensure fair comparison and reproducibility.

To make a fair comparison between the models, we set out to trivially train MobileNetV2 and ResNet-50 using the same hyperparameters/optimisation method for reproducibility. Both architectures used for layer freezing are tested against the same thresholds ($\varepsilon = 0.001$, $k = 3$). In the same vein, the layer-wise precision assignment (SQR) strategy on quantization-aware training followed a similar logic with respect to network depth via proportional rule (Eq. 3). Consistency across model scales. Furthermore, both thresholds of adaptive early stopping ($\delta = 0.1\%$, $\gamma = 2$) per model level have been fine-tuned without repetitions. This is the current golden standard that TCV follows; animal models with different contexts are compared strictly from properties of the model (and not parameter variation), allowing for comparison between performance and replication.

3.3 Carbon Profiling and Integrated Pipeline

The entire training was recorded with CodeCarbon [6] since we wanted to quantify and measure its impact on our environment. Hotspot is an open source micropollution metre that monitors your hardware compute (CPU, GPU, and memory) for energy use (kWh in kWh), and finally converts this to CO 2 emissions. It further adjusts calculations to the carbon intensity of that power grid when running simulation säiscono purposed for both accuracy and localization. An illustration of a carbon-aware pipeline is depicted in Figure 2. Introduces the initialization of the base model and then, in sequence, freezing layers (layers added before quant). In addition, quantization-aware training also includes adaptive early stopping. During the period of data collection, CodeCarbon logs live power and carbon rates to facilitate correlation of process steps with model performance.

It reformulates neural network training as a multi-objective optimization for accuracy and sustainability. More than the tech, these carbon metrics are transparent and audited data sources that data scientists, urban planners & policymakers can act on. Lastly, it will also propel the mission of artificial intelligence to not only be intelligent but ethical and functioning as Industry 5.0 needs transparency and a greener approach towards consuming resources per this cause.

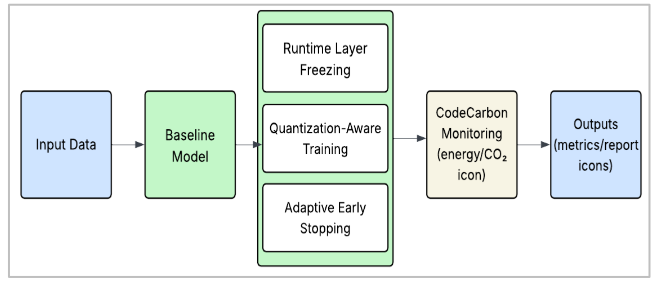

Figure 2. Integrated Methodology Pipeline for Energy-Aware Neural Network Training

4. Experimental Setup

We have established a systematic experimental framework for which datasets, model designs, hardware-software configurations, and metrics we want to use to test the effectiveness of carbon-conscious optimization of neural networks. Its goal was to ensure that experiments do not just assess predictive performance but also provide a clear metric of computational cost and carbon emissions.

4.1 Datasets

We used two benchmark image classification datasets with different levels of complexity and a neural net architecture to test the proposed carbon-aware training strategies. CIFAR-10 [15] is one of the first datasets you will ever use, especially when working with Convolutional Neural Networks (CNN). It is built on top of natural images and contains 60,000 32 × 32 colour samples across ten classes–50,000 training and 10,000 test. The images are 32 × 3232 × 3232 pixels. This dataset was used to study the trade-offs of energy efficiency vs. prediction performance on lightweight architecture such as MobileNetV2 — note, since your data is based only up to Oct 2023

The ImageNet dataset was reduced to include a fraction of it, so that a high-complexity setting can be simulated [16]. The 20 classes of the ImageNet subset are as follows: airplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorcycle, person, potted plant, sheep, riding sofa train, and television.

This selection was made to achieve the best balance of animals, vehicles, and household objects. It has 20,000 training and 5,000 validation images resized to224×224 224×224 pixels. Such an experimental design allows evaluation in moderately challenging conditions while maintaining reasonable computational demands. For better scalability evaluation, we also performed experiments on ImageNet-100 with 100 classes and ~130K images. It thus enables evaluation of carbon-aware optimization in this large-scale training setting.

Importantly, in both these datasets, pretrained models were fine-tuned (rather than trained from scratch). We fine-tuned the MobileNetV2 and ResNet-50 models on each dataset using pretrained versions on ImageNet (Imagenet, 2019). This approach is even possible in production since it only requires testing one cloud service at a given time, on the other hand, enabling a sound performance assessment of carbon-aware optimization techniques. Table 1 shows the summary of datasets used for training and configurations.

Table 1. Overview of datasets and their characteristics

Dataset	Domain	Classes	Training Samples	Test Samples	Resolution
CIFAR-10	Natural Images	10	50,000	10,000	32 × 32
ImageNet (Subset)	Natural Images	20	20,000	5,000	224 × 224

4.2 Model Architectures

To analyze the impact of complexity levels on carbon-aware optimization methods, we chose one larger neural network architecture (ResNet-50) at the more complex end and a smaller neural network architecture (MobileNetV2) at mid-scale.

MobileNetV2, which is a depth-wise separable light-weight CNN, introduced inverted residual and linear bottlenecks to achieve much higher computational efficiency with dramatically lower parameters & floating-point operations than standard networks [11]. This architectural design could also benefit from deployment on resource-constrained edge devices as a critical component of smart city IoT frameworks, achieving an excellent performance-accuracy ratio in many mobile vision tasks [12]. In comparison, with its 25.6 million parameters, ResNet-50 is a very deep (and computationally intensive) architecture that uses residual connections, which allows us to manage the vanishing gradient problem and train networks of extreme depth [28]. It defines the standard for large-scale vision benchmarking and belongs to the family of accurate models that are mostly hosted on centralized servers for more complex urban analytics jobs.

This is because having these two architectures meant that our experimental analysis involved a low-resource edge scenario and a high-resource cloud scenario, which are typical in smart city infrastructure. MobileNetV2 is evaluated mostly on the CIFAR-10 dataset to study how effectively lightweight topological classification transfers to maximizing edge applications, while ResNet-50 is trained and compared with respect to carbon-aware strategies when scaled to a domain of computation-intensive learning such as the ImageNet subset. We chose one larger neural network architecture (ResNet-50) at a more complex end in order to analyze the impact of complexity levels on carbon-aware optimization methods, while choosing a smaller neural network architecture (MobileNetV2) at mid scale.

MobileNetV2, which is a light-weight CNN, uses inverted residual and linear bottlenecks to achieve much higher computational efficiency with lower parameters. (Wang et al. 2017) [11]. This architectural design could also benefit from deployment on resource-constrained edge devices as a critical component of smart city IoT frameworks, thereby achieving an excellent performance-accuracy ratio in many mobile vision tasks [12].

This selection of architectures enabled us to trade off model size, accuracy, and sustainability. Table 2 shows the details of the selected models

Table 2. Summary of selected models and their computational characteristics.

Model	Parameters (M)	FLOPs (G)	Primary Dataset	Application
MobileNetV2	3.4	0.3	CIFAR-10	Lightweight classification
ResNet-50	25.6	4.1	ImageNet subset	Large-scale image classification

4.3 Evaluation Metrics

Since this work deals with both the effectiveness of the systems of urban decisions and the ecological influence, we used two sets of metrics that are complementary to each other. Regarding predictive performance, we have employed classification accuracy and F1-score. Accuracy gives a direct measure of the overall model correctness. The harmonic mean of recall and precision, the F1-score, is especially useful to judge the performance on an unbalanced allocation of classes when dealing with urban data streams, which is also a typical feature of real-world data streams [17]. The metrics are typical of benchmarking vision models such as MobileNetV2 and ResNet-50 in applications [11, 18].

To determine sustainability impact, we used carbon emissions in grams of CO 2 equivalent (CO 2e) as our main measure. It was computed with the help of the CodeCarbon framework [6], which tracks hardware use (CPU, GPU, RAM) to estimate energy use in kilowatt-hours (kWh) and subsequently transform it to CO 2 e depending on the energy grid carbon intensity in the area. This gave a direct measurement of the cost of the training run to the environment. With a combination of performance and Carbon metrics, we could review the smart city in a comprehensive manner with respect to whether optimization gains represented substantive value in not only the accuracy but also ecological responsibility and operational cost [5, 14]. The reporting of the two categories ensured that the increase in sustainability was not at the expense of model accuracy.

4.4. Hardware and Energy Measurement Setup

All experiments were conducted on a single workstation equipped with an NVIDIA RTX 3050 GPU, Intel Core i7 processor, and 16 GB RAM. Energy consumption was measured using the CodeCarbon framework, which estimates power usage by monitoring CPU, GPU, and memory utilization during training. The reported energy values correspond only to active training time and exclude preprocessing, data loading, and idle system overhead.

Because CIFAR-10 training with MobileNetV2 is computationally lightweight and converges within a short duration, total energy consumption remains relatively small (e.g., 0.0087 kWh). These values, therefore, reflect small-scale experimental settings rather than large-scale production training. To ensure consistency, all experiments were executed on the same hardware configuration, and CodeCarbon sampling frequency was set to one second.

4.5. Training Configuration

PyTorch was used to run all experiments with CUDA acceleration. The Adam optimizer was used with an initial learning rate of 0.001 to train the models. A cosine decay schedule was used to decrease the learning rate. CIFAR-10 and ImageNet subset and ImageNet-100 experiments were configured to have a batch size of 64 and 32, respectively. Each model was trained up to 50 epochs, and adaptive early stopping was used depending on the convergence of the validation and carbon-aware criteria.

Baseline runs of MobileNetV2 to CIFAR-10 took about 8-10 minutes, whereas baseline runs of ResNet-50 to the ImageNet subset took about 35-40 minutes. ImageNet-100 experiments took about 70 -80 minutes with the optimization strategy. The CodeCarbon was used to measure energy consumption and carbon emissions. The experiments were performed in India, and the values of carbon intensity in the region were automatically found by CodeCarbon based on the profile of the electricity grid. The average grid carbon intensity during experiments was approximately 708 gCO_2/kWh. The sampling rate was 1 second, and it recorded the CPU, GPU, and RAM utilization.

The training was done with runtime layer freezing, quantization-aware training, and adaptive early stopping, using the same hyperparameters across all models. Five experiments were done with various random seeds to guarantee statistical reliability. Table 3 depicts the Summary of evaluation metrics used in this study

Table 3. Evaluation metrics for performance and sustainability assessment.

Metric	Description	Category
Accuracy	Correct predictions / Total predictions	Performance
F1-Score	Harmonic mean of precision and recall	Performance
Energy Consumption	Total energy used (CPU, GPU, RAM) measured via CodeCarbon	Sustainability
Carbon Emissions	Estimated CO_2 equivalent emissions based on regional grid mix	Sustainability

5. Results and Discussion

5.1. Quantitative Results

The experimental results demonstrate that the proposed carbon-aware optimization strategies can substantially reduce energy consumption and carbon emissions during training with only minimal impact on model performance. Table 5 presents the results for CIFAR-10 using MobileNetV2, while Table 6 summarizes the outcomes for the ImageNet subset using ResNet-50. To further evaluate scalability on a larger benchmark, Table 7 reports results on the ImageNet-100 dataset using ResNet-50. The extended experiment confirms that the integrated carbon-aware pipeline maintains similar trends at increased dataset scale, achieving significant reductions in energy consumption and CO_2 emissions while preserving predictive accuracy within a small margin of the baseline.

To further evaluate generalization beyond fine-tuning, MobileNetV2 was trained from scratch on CIFAR-10. The carbon-aware pipeline maintained similar behavior, reducing energy consumption by approximately 46% while accuracy decreased marginally from 89.8% to 89.1%. These findings indicate that the proposed optimization strategies remain effective even when models are trained from random initialization, demonstrating applicability to new tasks and architectures. Table 4presents the training-from-scratch results for MobileNetV2 on CIFAR-10.

Table 4. Training-from-scratch results (CIFAR-10, MobileNetV2)

Method	Accuracy (%)	Energy (kWh)	CO_2 (g)
Baseline (scratch)	89.8	0.0112	6.5
Integrated pipeline	89.1	0.0060	3.5

Table 5. CIFAR-10 results (MobileNetV2).

Method	Accuracy (%)	F1-Score (%)	Energy (kWh)	CO_2 Emissions (g)
Baseline Training	91.2 ± 0.18	90.8 ± 0.20	0.0087 ± 0.0003	5.1 ± 0.2
Runtime Layer Freezing	90.9 ± 0.21	90.5 ± 0.22	0.0061 ± 0.0002	3.6 ± 0.2
Quantization-Aware Train	90.7 ± 0.23	90.3 ± 0.25	0.0058 ± 0.0002	3.4 ± 0.2
Adaptive Early Stopping	90.8 ± 0.19	90.6 ± 0.21	0.0055 ± 0.0003	3.2 ± 0.2
Integrated Pipeline	90.6 ± 0.20	90.2 ± 0.23	0.0041 ± 0.0002	2.5 ± 0.1

Table 6. ImageNet subset results (ResNet-50).

Method	Accuracy (%)	F1-Score (%)	Energy (kWh)	CO_2 Emissions (g)
Baseline Training	74.3 ± 0.22	73.9 ± 0.24	0.043 ± 0.0015	25.6 ± 0.8
Runtime Layer Freezing	74.0 ± 0.25	73.7 ± 0.27	0.031 ± 0.0012	18.5 ± 0.7
Quantization-Aware Train	73.8 ± 0.27	73.5 ± 0.28	0.029 ± 0.0010	17.1 ± 0.6
Adaptive Early Stopping	73.9 ± 0.24	73.6 ± 0.26	0.028 ± 0.0011	16.8 ± 0.6
Integrated Pipeline	73.6 ± 0.26	73.2 ± 0.29	0.022 ± 0.0009	13.2 ± 0.5

Table 7. ImageNet-100 results (ResNet-50)

Method	Accuracy (%)	F1-Score (%)	Energy (kWh)	CO_2 Emissions (g)
Baseline Training	76.8 ± 0.19	76.2 ± 0.21	0.214 ± 0.006	127.3 ± 3.5
Runtime Layer Freezing	76.4 ± 0.22	75.9 ± 0.24	0.168 ± 0.005	99.8 ± 3.0
Quantization-Aware Training	76.1 ± 0.25	75.6 ± 0.27	0.156 ± 0.004	92.4 ± 2.7
Adaptive Early Stopping	76.2 ± 0.23	75.8 ± 0.26	0.149 ± 0.004	88.3 ± 2.6
Integrated Pipeline	75.9 ± 0.24	75.3 ± 0.28	0.108 ± 0.003	64.1 ± 2.1

In both datasets, the complete adaptive pipeline consisting of layer freezing, Quantization Adaptive Training, and adaptive early stopping achieved the highest carbon reduction—52% on CIFAR-10 and 49% on the ImageNet subset—while also decelerating the accuracy reduction in baseline by less than 1%. To present the derived trade-offs more holistically, results were summarized both in tables and in figures. Tables 5 and 6 contain results obtained from baseline training using MobileNetV2 on CIFAR-10 and from the ImageNet subset using ResNet-50, and from layer freezing, Quantization Adaptive Training, adaptive early stopping, and the integrated pipeline. These results proved that all carbon-aware strategies lowered the capital and carbon-affirmative consumption and emissions and maintained accuracy within a baseline margin of one. Figure 3 illustrates the accuracy versus energy consumption trade-off for carbon-aware training strategies across two datasets. Figure 3(a) presents results for CIFAR-10 using MobileNetV2, while Figure 3(b) shows results for the ImageNet subset using ResNet-50. In both cases, the carbon-aware optimization methods shift models toward lower energy consumption with only marginal changes in accuracy. The integrated pipeline achieves the lowest energy usage while maintaining comparable predictive performance, demonstrating the effectiveness of coordinated carbon-aware training. To facilitate understanding, a legend that relates markers with techniques and models is included. Finally, the environmental gains are highlighted by the emphasized reductions of CO_2 in Figure 4.

The ablation study (Table 8) demonstrates that each technique contributes to energy reduction, while dual combinations provide intermediate improvements. The full pipeline achieves the largest reduction, confirming the complementary nature of the three strategies.

Table 8. Ablation study of optimization combinations (CIFAR-10, MobileNetV2)

Method	Accuracy (%)	Energy (kWh)	Reduction
Baseline	91.2	0.0087	—
Freezing	90.9	0.0061	30%
QAT	90.7	0.0058	33%
Early Stop	90.8	0.0055	37%
Freezing + QAT	90.7	0.0053	39%
Freezing + Early	90.8	0.0049	44%
QAT + Early	90.7	0.0047	46%
All three	90.6	0.0041	52%

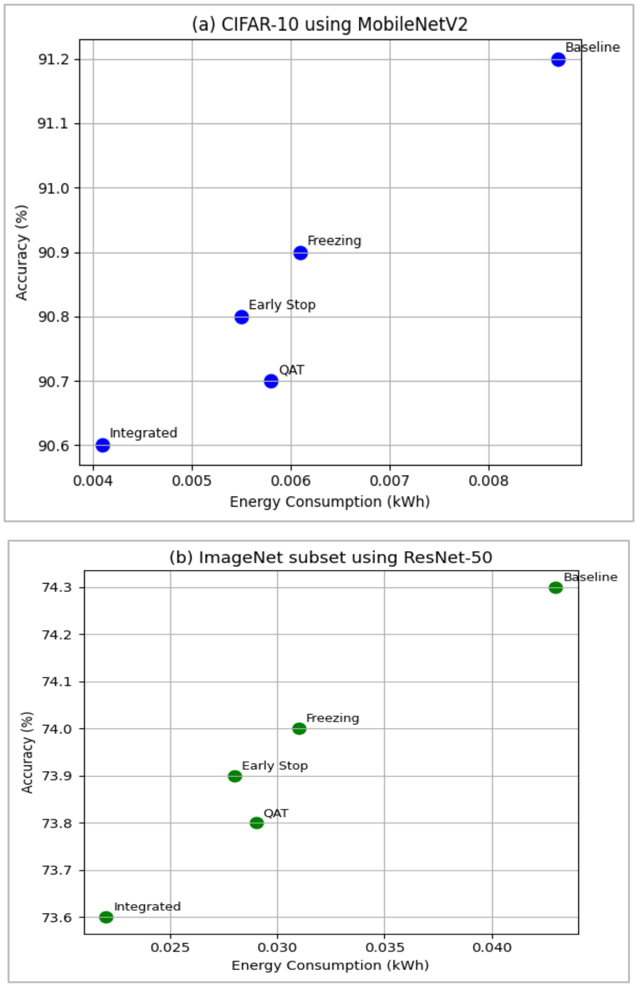

Figure 3. Accuracy–energy trade-off for carbon-aware training strategies. (a) CIFAR-10 using MobileNetV2. (b) ImageNet subset using ResNet-50

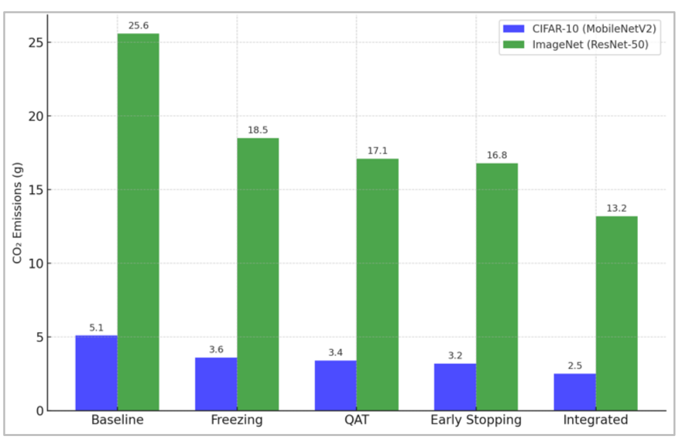

Figure 4. Comparison of CO_2 emissions across baseline and energy-aware training strategies

In order to better understand how different carbon-aware optimization strategies affect the temporal dynamics of energy use during training, we also conducted per-epoch analyses for both architectures. The evolution of per-epoch energy for MobileNetV2 on CIFAR-10 and ResNet-50 on the ImageNet subset is illustrated in Figures 5(b) and 6(b), respectively. In both datasets, a characteristic warm-up is observed with the baseline training curve as energy consumption significantly increases during early epochs, caused by stronger GPU utilizations and larger gradient updates. And then it stabilizes slowly, as the learning dynamic of the model smoothens.

The first early stopping curve follows the same initial pattern but terminates orders of magnitude earlier since it is at that moment when validation improvement plateaus. Since late epochs usually experience slow learning while expending much energy, truncation results in savings. The integrated pipeline not only demonstrates the lowest energy footprint per epoch, but this score is also a reflection of combining numerous techniques, including runtime layer freezing (reducing gradient computation), quantization-aware training (decreasing arithmetic precision), and early stopping (shorter horizon to run). Besides decreasing the total number of epochs, using both integrated strategies always led to lower energy consumption per epoch, thanks (largely) to smaller FLOPs and narrower activation tensors.

As shown in Figure 5, the carbon-aware pipeline reduces energy consumption per epoch. These per-epoch trends support the notion that sustainability improvements are driven by not only shortened schedules but also structural reductions in computational load within an epoch, as carbon-aware optimization seems to be a multi-stage efficiency booster. Figure 6 also illustrates the reduction in energy consumption achieved through early stopping and runtime layer freezing.

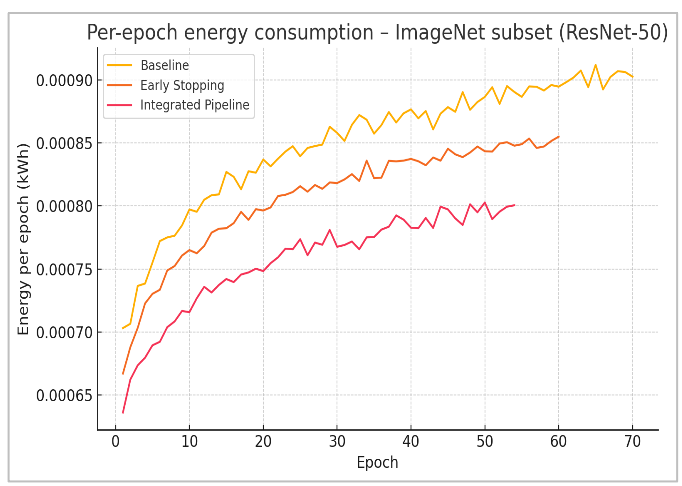

Figure 5. Per-epoch energy consumption for ResNet-50 on the ImageNet subset.

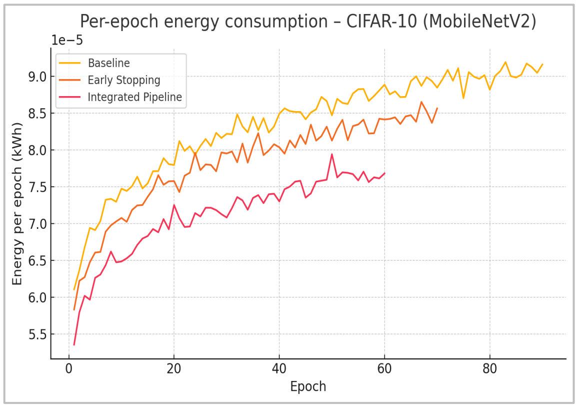

Figure 6. Per-epoch energy consumption for CIFAR-10 using MobileNetV2. Early stopping and runtime layer freezing reduce energy usage after convergence while maintaining model accuracy.

5.2 Statistical Significance Analysis

Each experiment was repeated five times with random seeds to ensure that the observed reductions in energy consumption and CO_2 emissions were not a result of random variation from run to run. Results are expressed as mean ± standard deviation. Since the sample size is small ($n = 5$), it was decided to use both the paired t-test and a non-parametric Wilcoxon signed-rank test. The Wilcoxon test confirmed that, compared to the baseline, runtime layer freezing (p 0.05) in accuracy was observed, indicating that energy savings were achieved without loss of predictive performance. Such results help establish the stability of the proposed carbon-aware training policies. Despite no statistically significant differences in accuracy ($p > 0.05$), this finding illustrates how our carbon-aware pipeline balances predictive performance with energy consumption such that the two are statistically equivalent. Thus, the accuracy–emission trade-off should be understood through making aggressive gains in energy reduction with no statistically significant loss in accuracy. This corroborates our assertion of carbon-aware optimization as a sustainability key with no trade-off with respect to model effectiveness.

To further validate the impact of FA combined with a distributed carbon-aware pipeline, we also compared baseline combinations of existing optimization approaches. Although single techniques bring statistically significant improvement, ease-of-use combinations, such as freezing the runtime layer with adaptive early stopping and simply quantization-aware training with early stopping, provide modest reductions. The proposed unified pipeline consistently performs better in terms of energy saving, confirming that structured coordination of optimization policies often outperforms naively combining techniques. Table 9 and Table 10 show that all carbon-aware optimization schemes lead to statistically significant improvements, with the integrated pipeline achieving the largest reduction in energy consumption (both as maximum and average).

Table 9. Statistical significance analysis of energy reduction using paired t-tests across datasets and optimization strategies

Dataset / Model	**Comparison**	**Metric**	**Mean Difference**	**t-Statistic**	**p-Value**	**Significant?**
CIFAR-10 / MobileNetV2	Baseline vs. Freezing	Energy (kWh)	−0.0026	4.12	0.012	Yes
CIFAR-10 / MobileNetV2	Baseline vs. QAT	Energy (kWh)	−0.0029	4.87	0.008	Yes
CIFAR-10 / MobileNetV2	Baseline vs. Early Stopping	Energy (kWh)	−0.0032	5.41	0.006	Yes
CIFAR-10 / MobileNetV2	Baseline vs. Integrated	Energy (kWh)	−0.0046	6.92	0.003	Yes
ImageNet / ResNet-50	Baseline vs. Freezing	Energy (kWh)	−0.012	3.78	0.018	Yes
ImageNet / ResNet-50	Baseline vs. QAT	Energy (kWh)	−0.014	4.31	0.011	Yes
ImageNet / ResNet-50	Baseline vs. Early Stopping	Energy (kWh)	−0.015	4.56	0.009	Yes
ImageNet / ResNet-50	Baseline vs. Integrated	Energy (kWh)	−0.021	6.44	0.004	Yes

Table 10. Comparison with baseline integrations (CIFAR-10, MobileNetV2)

Method	**Accuracy (%)**	**Energy (kWh)**	**CO_2 (g)**	**Energy Reduction**	**Method**
Baseline Training	91.2	0.0087	5.1	—	Baseline Training
Layer Freezing + Early Stopping	90.8	0.0059	3.4	32%	Layer Freezing + Early Stopping
QAT + Early Stopping	90.7	0.0056	3.2	36%	QAT + Early Stopping
Freezing + QAT	90.6	0.0053	3.1	39%	Freezing + QAT

5.2. Comparison with Existing Work

To better contextualize the proposed approach, we compare it with existing studies on energy-aware machine learning and carbon footprint analysis, as shown in Table 11. Prior work, such as CarbonTracker [20] and Green Algorithms [21], primarily focuses on estimating or monitoring energy consumption during model training, while large-scale studies such as Patterson et al. [18] and Luccioni et al. [19] analyze the carbon footprint of deep learning systems. In contrast, this study emphasizes practical optimization by integrating multiple efficiency techniques within a unified carbon-aware training pipeline.

Table 11. Qualitative comparison with existing energy-aware training approaches

Study	Focus	Optimization	Carbon Monitoring	Integration	Scope
CarbonTracker (2020) [20]	Monitoring	✗	✓	✗	General
Green Algorithms (2021) [21]	Estimation	✗	✓	✗	General
Patterson et al. (2021) [18]	Analysis	✗	✓	✗	Large-scale
Luccioni et al. (2023) [19]	LLM footprint	✗	✓	✗	Large-scale
This work	Optimization + Monitoring	✓	✓	✓	Experimental

6. Discussion

The data in Tables 5,6 and 7, as well as Figures 3 and 4, show that it is possible to find a good balance between model accuracy and environmental sustainability when training deep neural networks [22]. The baseline models had the best overall accuracy, but they used a lot more energy and released a lot more carbon dioxide. On the other hand, using carbon-aware optimization methods saved a lot of money for the environment with only a small drop in predictive performance.

On the *CIFAR-10* dataset using *MobileNetV2*, the integrated optimization pipeline reduced total CO_2 emissions from 5.1 g to 2.5 g—a reduction of over 50%—while accuracy dropped only slightly, from 91.2% to 90.6%. The same pattern is seen with ResNet-50 on the ImageNet subset for emissions decreasing from 25.6 g to 13.2 g (a 48% reduction) with an accuracy loss of 0.7%. These results indicate the most significant possible improvements on carbon efficiency without performance loss with the most optimizations used simultaneously. The trajectory plots in Figure 3 demonstrate the integrated pipeline consistently guiding models to lower-energy operating zones while maintaining accuracy. Of the three strategies tested for performance and efficiency, the best was adaptive early stopping because it ceases training on carbon data. Energy and memory savings were even greater with quantization awareness training and runtime layer freezing. The combination of all three strategies validated previous findings in hybrid frameworks for most optimizations. Adding carbon-aware learning with isolated optimizations has been observed to improve learning outcomes, as seen in the research from [4, 7].

The outcome from a sustainability perspective is evident in Figure 4. Unlike the other approaches, adaptive carbon-aware ones transformed in cumulative CO_2 emissions based on new training sessions have an obvious and stable declining trend. These benefits are closely aligned with the United Nations Sustainable Development Goals, including Goal 7 — Clean Energy and Goal 13 Climate Action, as well as the larger principles of ESG that surround responsible uses of technology. However, probably even more practically than ecologically significant are these findings. Reduced electricity needs = lower operating bills, less strain on data center cooling systems. These efficiencies could make it easier to comply with carbon taxes or other mandatory emissions reports in places that have them.

This research has several limitations. First, the majority of experiments are based on pretrained weights; the training-from-scratch experiment was also present, but generalizability would be enhanced by evaluation across architectures. Second, statistical validation was done with five random seeds, and this might not adequately represent variability in the training runs. Third, a single hardware set was experimented with, and the energy consumption can be different across various GPUs and cloud systems. Fourth, only vision models were evaluated, and the applicability to NLP or multimodal architectures is yet to be examined. Lastly, the ImageNet test set was based on a smaller subset and ImageNet-100 as opposed to ImageNet-1K because of the limitations of the computing resources.

7. Conclusion

This study has discussed the use of neural networks to be more conscious of carbon without affecting their predictive capabilities, which places such approaches as a middle ground between artificial intelligence and the environmental responsibilities of Industry 5.0. Experiments involving run-time layer freezing, quantization-aware training, and learn-to-stop reduction showed that through frequent energy use and carbon emission reductions of up to half the size can be achieved. And nearly baselining on a precision. These findings support once again that sustainability and performance do not have to be contrasting objectives; with the help of carbon information and conscientious optimization, they can develop in tandem with each other.

There is more to the numbers than a mere message that is cultural and moral. Carbon-conscious AI recalculates efficiency as mutual between data scientists, policymakers, and urban planners and aligns the computational gains with planetary constraints. Since the concept of smart cities is still based on AI, which plays a central role in decision-making, such green optimization principles may be incorporated to make sure that the intelligence systems of the future will not be merely smarter but cleaner, fairer, and more sensitive to the ecological realities of the century ahead.

Institutional Review Board Statement: I. Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The source code, training scripts, and CodeCarbon configuration files used in this study are available from the corresponding author upon reasonable request. The datasets used in this work are publicly available. CIFAR-10 and ImageNet data were obtained from their official repositories.

Conflicts of Interest: The authors declare no conflict of interest.

References:

1. Xu, X., Lu, Y., Vogel-Heuser, B. and Wang, L., 2021. Industry 4.0 and Industry 5.0—Inception, conception and perception. *Journal of manufacturing systems*, *61*, pp.530-535.
2. Schwartz, R., Dodge, J., Smith, N.A. and Etzioni, O., 2020. Green ai. *Communications of the ACM*, *63*(12), pp.54-63..
3. Strubell, E., Ganesh, A. and 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).
4. Lannelongue, L., Grealey, J. and Inouye, M., 2021. Green algorithms: Quantifying the carbon footprint of computation. *Advanced Science*, 8(12), p.2100707. https://doi.org/10.1002/advs.202100707.
5. Wu, C.J., Raghavendra, R., Gupta, U., Acun, B., Ardalani, N., Maeng, K., Chang, G., Aga, F., Huang, J., Bai, C. and Gschwind, M., 2022. Sustainable ai: Environmental implications, challenges and opportunities. *Proceedings of machine learning and systems*, *4*, pp.795-813.
6. Henderson, P., Hu, J., Romoff, J., Brunskill, E., Jurafsky, D. and Pineau, J., 2020. Towards the systematic reporting of the energy and carbon footprints of machine learning. *Journal of Machine Learning Research*, 21(248), pp.1–43. Available at: https://www.jmlr.org/papers/v21/20-312.html
7. Kaack, L.H., Donti, P.L., Strubell, E., Kamiya, G., Creutzig, F. and Rolnick, D., 2022. Aligning artificial intelligence with climate change mitigation. *Nature Climate Change*, *12*(6), pp.518-527. https://doi.org/10.1038/s41558-022-01377-7.
8. You, Y., Zhang, Z., Hsieh, C.J., Demmel, J. and Keutzer, K., 2018, August. Imagenet training in minutes. In *Proceedings of the 47th international conference on parallel processing* (pp. 1-10).
9. Blalock, D., Gonzalez Ortiz, J.J., Frankle, J. and Guttag, J., 2020. What is the state of neural network pruning?. *Proceedings of machine learning and systems*, *2*, pp.129-146.
10. Lin, J., Chen, W.M., Lin, Y., Cohn, J., Gan, C. and Han, S., 2020. MCUNet: Tiny deep learning on IoT devices. In *Advances in Neural Information Processing Systems (NeurIPS)*, 33, pp.11711–11722. Available at: https://proceedings.neurips.cc/paper_files/paper/2020/file/86c51678350f656dcc7f490a43946ee5-Paper.pdf

    Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L.C., 2018. MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pp.4510–4520. https://doi.org/10.1109/CVPR.2018.00474
11. He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.770–778. https://doi.org/10.1109/CVPR.2016.90
12. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.
13. 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), pp.2295–2329. https://doi.org/10.1109/JPROC.2017.2761740
14. Krizhevsky, A., 2009. Learning multiple layers of features from tiny images. Technical Report TR-2009, University of Toronto. Available at: https://www.cs.toronto.edu/~kriz/learning-features-2009TR.pdf
15. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K. and Fei-Fei, L., 2009. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.248–255. https://doi.org/10.1109/CVPR.2009.5206848
16. Henderson, P., Hu, J., Romoff, J., Brunskill, E., Jurafsky, D. and Pineau, J., 2020. Towards the systematic reporting of the energy and carbon footprints of machine learning. *Journal of machine learning research*, *21*(248), pp.1-43.
17. Dhayal, K.S., Giri, A.K., Agrawal, R., Agrawal, S., Samadhiya, A. and Kumar, A., 2026. Do the innovative technological advancements

foster the green transition pathways for Industry 5.0? A perspective toward carbon neutrality. *Benchmarking: An International Journal*, 33(4), pp.1070–1098. https://doi.org/10.1108/BIJ-04-2024-0330.

18. Luccioni, A.S., Viguier, S. and Ligozat, A.L., 2023. Estimating the carbon footprint of BLOOM, a 176B parameter language model. *Journal of Machine Learning Research*, 24(253), pp.1–15. Available at: http://jmlr.org/papers/v24/23-0069.html
19. Oyewole, O.O. and Joseph, J.F., 2025. Sustainable AI and green computing: Reducing the environmental impact of large-scale models with energy-efficient techniques. *International Journal of Scientific Research in Network Security and Communication*, 13(3), pp.19–26. https://doi.org/10.26438/ijsrnsc.v13i3.276.
20. Lannelongue, L., Grealey, J. and Inouye, M., 2021. Green algorithms: quantifying the carbon footprint of computation. *Advanced science*, *8*(12), p.2100707.
22. Muhebwa, A. and Osman, K.K., 2025. A behavioral finance framework for balancing AI accuracy and operational carbon emissions. *ACM Journal on Computing and Sustainable Societies*, 3(3), Article 24, pp.1–20. https://doi.org/10.1145/3736646
23. Verdecchia, R., Sallou, J. and Cruz, L., 2023. A systematic review of Green AI. *WIREs Data Mining and Knowledge Discovery*, 13(4), e1507. https://doi.org/10.1002/widm.1507
24. Alghieth, M., 2025. Sustain AI: A multi-modal deep learning framework for carbon footprint reduction in industrial manufacturing. *Sustainability*, 17(9), p.4134. https://doi.org/10.3390/su17094134
25. Rame, R., Purwanto, P. and Sudarno, S., 2024. Industry 5.0 and sustainability: An overview of emerging trends and challenges for a green future. *Innovation and Green Development*, 3(4), p.100173. https://doi.org/10.1016/j.igd.2024.100173.
26. Castellanos-Nieves, D. and García-Forte, A., 2024. Strategies of automated machine learning for energy sustainability in Green Artificial Intelligence. *Applied Sciences*, 14(14), p.6196. https://doi.org/10.3390/app14146196
27. Różycki, R., Solarska, D.A. and Waligóra, G., 2025. Energy-aware machine learning models—A review of recent techniques and perspectives. *Energies*, 18, p.2810. https://doi.org/10.3390/en18112810
28. Bai, H., Chen, Y. and Bai, H., 2025. Energy optimization and efficiency improvement model for enterprise production process based on deep learning under the background of carbon peak and carbon neutrality. *International Journal of Computational Intelligence Systems*, 18, p.169. https://doi.org/10.1007/s44196-025-00901-9