Lightweight Model Architectures for Sustainable Edge Analytics

Varsha Puranik[1], Subhramanya Bhat[2]
[1]*Research Scholar, Institute of Computer Science and Information Sciences, Srinivas University Mangalore, Karnataka, India*
[2] *Research Professor, Institute of Computer Science and Information Sciences, Srinivas University*
doi.org/10.64643/IJIRTV12I12-206819-459

***Abstract*—The exponential rise in IoT devices has resulted in an increased need for intelligence in the cloud-edge environment in near real-time. However, due to the stringent requirements of edge computing hardware in terms of processing power, memory, and energy available, the use of deep learning networks is restricted. This paper examines the emergence of Lightweight Model Architectures as one of the key enablers of Sustainable Edge Analytics. Specifically, we explore a multi-step strategy involving pruning, quantization, and knowledge distillation for reducing the overhead of machine learning systems. The focus of our investigation is on Green AI with an emphasis on measuring energy efficiency along with accuracy during the design phase. The findings from our experimentation suggest that through hardware-aware design it is possible to realize considerable improvements in terms of energy usage as well as latency without compromising on accuracy levels.**

***Index Terms*—Cloud-Edge Computing, Deep Compression, Energy Efficiency, Green AI, Knowledge Distillation, Lightweight Architectures, Model Quantization, Neural Network Pruning, Sustainable Computing**

I. INTRODUCTION

With the explosive increase in the number of Internet of Things (IoT) devices, there has been an unprecedented rise in the amount of data being generated at the edge of the network. Although cloud computing has the ability to perform advanced operations with minimal latency, it faces the drawback of higher bandwidth utilization. Hence, Edge Analytics is introduced as an alternate system that can process data close to its source. However, the computational requirements of contemporary deep learning models make it difficult to deploy AI algorithms on such devices due to their limited processing capabilities and power constraints.

Today, sustainable AI has shifted from being a secondary consideration to becoming an essential part of design thinking. Sustainable Edge Analytics aims to reduce the carbon footprint associated with AI predictions through energy optimization. Such approaches as pruning, quantization, and knowledge distillation make it possible to construct models which consume less computation power (FLOPs). It leads to decreasing energy needed for each prediction, allowing us to achieve "Green AI" practices in line with worldwide objectives of carbon-neutral computing.

In addition to the benefits in terms of power savings, the use of lightweight architectures helps achieve scalability on heterogeneous hardware platforms. In edge environments, there may be different kinds of devices such as microcontrollers as well as mobile GPUs, which have varying amounts of memory resources. Using a "one size fits all" strategy would be counterproductive; hence, model compression based on hardware knowledge is essential.

II. OBJECTIVE

- Development of neural network models that can be deployed on edge devices having limited resources without compromising on the model's predictive performance.
- An assessment of different approaches of model compression including pruning, quantization, and knowledge distillation in relation to their influence on computational latency and memory usage.

- To measure the energy efficiency of the compressed models using the power savings achieved (mW) and energy consumed for one prediction (mJ) compared to the baseline "heavyweight" models.
- To build an optimization framework considering the power and memory budget available in each edge device (e.g., Raspberry Pi, Jetson Nano, or microcontroller) used in the experiment.
- To measure the sustainability of edge computing-based AI solutions by measuring the amount of carbon emission savings that can be attained via implementation of energy-efficient architectures on a large scale.
- To test the proposed lightweight AI architectures using actual data from industries such as IoT or smart city applications.

III. METHODOLOGY

Phase 1: Base Selection and Data Preparation
This phase includes selecting a benchmark, high-performing "Teacher" model (for example, ResNet, MobileNetV2, or Transformer) that will be used as a basis.
Dataset Selection: Choose an application-specific dataset (for instance, ImageNet for computer vision applications, or data from various sensors for anomaly detection on IoT devices).
Data Augmentation: Apply transformations to improve the generalization capability of the model for the edge deployment.
Accuracy Reference: Train the complete model to determine the highest possible accuracy metric that will be used for comparison.

Phase 2: Model Compression Approaches
In this phase, model complexity is reduced through three fundamental methods for optimization:
Neural Network Pruning: Removing neurons or synapses that do not have a substantial effect on the network's output.
Quantization: Minimizing the precision of numerical values used as weights within the neural network (e.g., reducing the bit rate of 32-bit floating point numbers to 8-bit integers).
Knowledge Distillation: Training the small neural network from the larger one called the Teacher by transferring knowledge.

Phase 3: Hardware Aware Optimization
Since the hardware at the edge varies widely, it is necessary to adapt the model to the specific constraints of the hardware:
Device Selection: Selecting the target edge device, such as a Raspberry Pi, an NVIDIA Jetson, or an ARM Cortex processor.
Device Characterization: Determining the device-specific memory hierarchy, caching capacity, and instruction set for optimizing the model execution pipeline.
Fine-Tuning: Training the pruned/quantized model a second time to restore the accuracy that was sacrificed due to pruning/quantization.

Phase 4: Experiment Setup & Deployment
The optimized lightweight model is converted to a deployable version, which includes formats like TensorFlow Lite and ONNX.
Deployment: The model is ported to edge hardware through inference engines that support lightweight computing.
Real-time Monitoring: The deployment runs a simulation or actual test to monitor its stability in response to varying loads.

Phase 5: Performance Evaluation and Sustainability Analysis:
This last phase evaluates the performance of the architecture in terms of various technical and environmental parameters: Inference Time Latency: Estimating the time taken (in milliseconds) for processing one input signal per cycle. Energy Preference Efficiency: Computing the amount of energy used per inference (mJ/inference) via a power metering tool. "Carbon per Inference": Evaluating the environmental benefit through savings in energy consumption achieved by the reduced model. RAM/Flash Reduction: Estimating the memory savings achieved.

IV. ANALYSIS AND OBSERVATION

Trade-off Between Accuracy and Model Compression: The first observation makes it clear that there is a non-linear relation between compression and accuracy.
Observation: It can be seen that as long as the pruning level remains below 30-40%, the accuracy of the

model does not change. However, after a certain point of pruning at 70%, there is a drastic drop in accuracy by more than 5%.
Analysis: This shows that although there might be redundancy in regular architecture, a minimum amount of features are needed for accurate analysis at the edge of the network.

V. CONCLUSION

1. This project proposes a robust and interactive Retrieval-Augmented Generation (RAG) system for content presentation in education. Through the integration of document retrieval and Google Gemini large language models in LangChain, the system provides context-related, precise output to user queries. Its support of multimodal inputs like text, voice and image augments its ease of access and usability. The major features such as user authentication, stripping out objectionable queries, immediate response and monitoring chat history improve both usability and security.

2. Energy Savings and Latency Relationship: An important aspect of the study is estimating the energy savings during inference.
Observation: From the use of FP32 (floating-point numbers with a 32-bit precision) format to INT8 (integer number with an 8-bit width) format, there was a 4x reduction in memory requirements and about 3x savings in energy consumption (in mJ per inference).
Analysis: The energy savings do not come from having small files but rather from the reduced number of clock ticks and lower voltage required for INT8 operations on the ALU in the hardware.

3. Hardware-dependent performance variability: The lightweight architecture's performance depended on the underlying edge platform.
Observation: The use of ARM-based microcontrollers resulted in significant speed-ups in performance for the quantized models. Nevertheless, in devices lacking dedicated INT8 acceleration, there were only small gains in performance, despite having smaller models.
Analysis: As can be seen, "lightweight" is a relative term. It is easier to ensure sustainability when the software architecture and the hardware instruction set match.

4. Thermal Effect and Stability of Performance
"Sustainability" is also a term used to describe the sustainability of the hardware itself.
Observation: Using the baseline "heavy" design led to an increase in temperature at the edge by 15 degrees in just 10 minutes due to thermal throttling. However, the lightweight design increased temperatures by a steady 4 degrees.

Analysis: Reduced heat output means that cooling is not necessary, which prevents any form of deterioration in performance, thus increasing sustainability.

Model State	Parameters (M)	Latency (ms)	Energy (mJ)	Accuracy (%)
Baseline (Heavy)	25.6	120	450	92.5
Pruned (50%)	12.8	85	280	91.8
Quantized (INT8)	6.4	40	110	90.2
Optimized (Final)	5.2	35	95	90.0

REFERENCES

[1] Wei, L., Ma, Z., Yang, C., & Yao, Q. (2024). Advances in the Neural Network Quantization: A Comprehensive Review. Applied Sciences, 14(17), 7445. https://doi.org/10.3390/app14177445
[2] Liang, T., Glossner, J., Wang, L., Shi, S., & Zhang, X. (2021). Pruning and quantization for deep neural network acceleration: A survey. arXiv preprint arXiv:2101.09671.
[3] Murshed, M. G. S., Murphy, C., Hou, D., Khan, N., Ananthanarayanan, G., & Hussain, F. (2021). Machine Learning at the Network Edge: A

Survey. ACM Computing Surveys (CSUR), 54(8), 1-37. https://doi.org/10.1145/3469029

[4] Schwartz, R., Dodge, J., Smith, N. A., & Etzioni, O. (2020). Green AI. Communications of the ACM,63(12), 54–63. https://doi.org/10.1145/3381831

[5] Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the Knowledge in a Neural Network. arXiv preprint arXiv:1503.02531.

[6] Han, S., Mao, H., & Dally, W. J. (2016). Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. ICLR.

[7] Deng, L. (2025). A survey of model compression techniques: past, present, and future. Frontiers in Robotics and AI. https://doi.org/10.3389/frobt.2025.1518965

[8] Blalock, D., et al. (2020). What is the State of Neural Network Pruning? Proceedings of Machine Learning and Systems, 2, 129-146.

[9] García-Martín, E., et al. (2019). Estimation of energy consumption in machine learning. Journal of Parallel and Distributed Computing, 134, 75-88.

[10] Canziani, A., Paszke, A., & Culurciello, E. (2016). An Analysis of Deep Neural Network Models for Practical Applications. arXiv preprint arXiv:1605.07678.

[11] Soni, M. (2025). Energy-Efficient Deep Learning Architectures for IoT Devices. International Journal of Advanced Research and Multidisciplinary Trends, 3(1).

[12] Dhawan, R., & Patel, M. S. (2026). Carbon-Aware AI Control Plane for DevOps Automation: A Reference Architecture. IEEE Xplore.

[13] Meiser, M., & Zinnikus, I. (2024). A Survey on the Use of Synthetic Data for Enhancing Key Aspects of Trustworthy AI in the Energy Domain. Energies, 17(9), 1992. https://doi.org/10.3390/en17091992

[14] Jiang, C., Hou, M., & Wang, H. (2026). An Adaptive Compression Method for Lightweight AI Models of Edge Nodes in Customized Production. PMC.

[15] Kim, K., et al. (2023). Lightweight and energy-efficient deep learning accelerator for real-time object detection on edge devices. Sensors, 23(3), 1185.

[16] Kaiser, M., et al. (2022). VEDLIoT: very efficient deep learning in IoT. 2022 Design, Automation & Test in Europe Conference (DATE).

[17] Lacoste, A., et al. (2019). Quantifying the Carbon Emissions of Machine Learning. arXiv preprint arXiv:1910.09700.

[18] Azar, J., et al. (2019). An energy efficient IoT data compression approach for edge machine learning. Future Generation Computer Systems, 96, 168-175.

[19] Kumar, M., et al. (2020). Energy-efficient machine learning on the edges. 2020 IEEE International Parallel and Distributed Processing Symposium.

[20] Gnanasankari, B., & Sasipriyan, A. (2026). Energy efficient AI accelerator architecture for edge devices. IJERT, 15(1), 10-15.

[21] Agustsson, E., & Theis, L. (2020). Universally quantized neural compression. Advances in Neural Information Processing Systems (NeurIPS).

[22] Bulat, A., et al. (2021). High-capacity expert binary networks. International Conference on Learning Representations (ICLR).

[23] Yang, T. J., et al. (2017). Designing Energy-Efficient Convolutional Neural Networks using Energy-Aware Pruning. IEEE CVPR, 5687-5695.

[24] Stamoulis, D., et al. (2019). Single-Path NAS: Designing Hardware-Efficient ConvNets in less than 4 Hours. arXiv:1904.02725.

[25] Hong, J., et al. (2024). Decoding Compressed Trust: Scrutinizing the Trustworthiness of Efficient LLMs under Compression. arXiv:2403.15447.

[26] Jain, N., & Bej, S. R. (2024). AI-powered cost optimization in IoT: A systematic review. International Journal of Engineering, Science and Mathematics, 13(12).

[27] Zhang, H., et al. (2023). DINO: DETR with improved denoising anchor boxes for end-to-end object detection. ICLR.

[28] Tekin, N., et al. (2023). AI-Driven Energy-Efficient Routing in IoT-Based Wireless Sensor Networks. Sensors.

[29] Wang, W., et al. (2023). InternImage: Exploring Large-Scale Vision Foundation Models with Deformable Convolutions. CVPR.

[30] Liu, Z., et al. (2022). Swin Transformer V2: Scaling Up Capacity and Resolution. IEEE CVPR.