Profiling Analysis of ResNet-18 & EfficientNet: A Green Software Perspective

Name: Ashu Jha
Mentor: Caterina Doglioni

Environmental Impact of AI

- **High Energy Demand**: Large-scale AI models (e.g., GPT-3) require massive computational resources.
- **Carbon Emission**s: GPT-3 training emitted 552 tons CO_2 comparable to 123 cars driven for one year
- **Water Usage**: GPT-3 training consumed 700,000 liters of freshwater enough for producing hundreds of vehicles.
- **Broader Adoption:** With billions of interactions monthly, AI's energy consumption is rapidly increasing.
- **Our Contribution:** Profiling resource use of image classifiers (ResNet, EfficientNet) to evaluate their sustainability, helping reduce AI's environmental footprint.

Why I Chose Image Classification for Analysis?

- Image classification models are widely used in AI and require high computational power, making them ideal for energy efficiency analysis.
- Different models like ResNet and EfficientNet, vary in complexity and energy use allowing us to compare efficiency trade-offs
- Optimizing image classification models reduces AI’s environmental impact and supports more sustainable applications in fields like medical imaging and autonomous systems.

Methodology: Training Setup & Energy Measurement

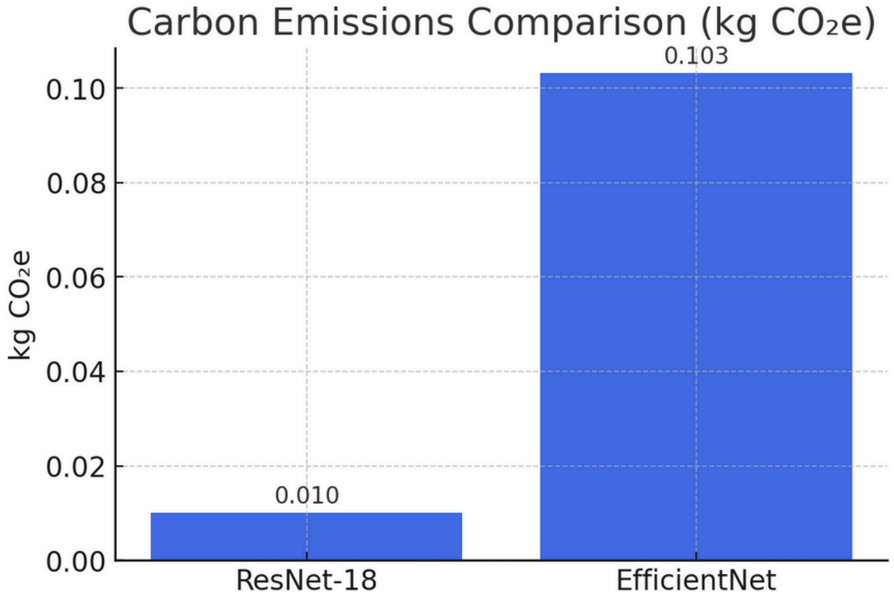
Training Setup

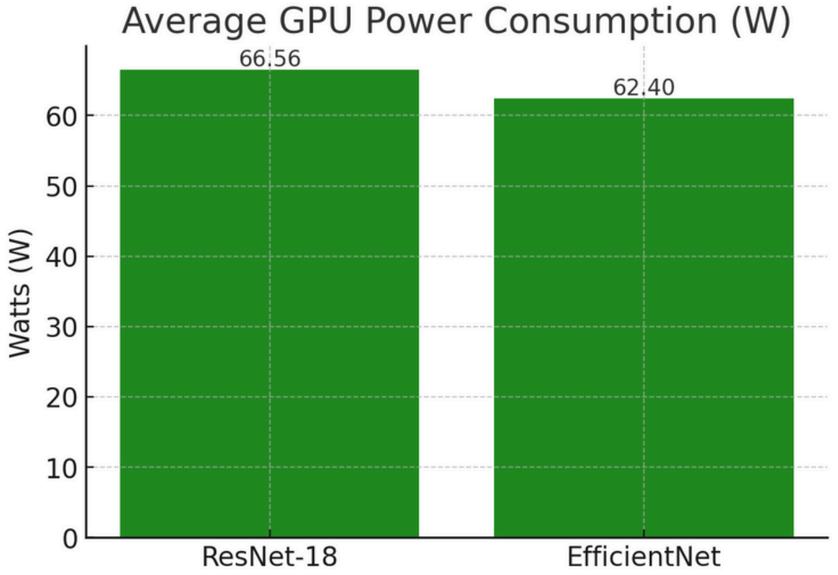
- GPU Used: Kaggle's cloud environment with 2× NVIDIA T4 GPUs
- CPU & RAM: Kaggle provides Intel Xeon CPU, 30GB RAM
- Storage: Cloud-based, auto-managed
- Dataset used: CIFAR 100
- Deep Learning Framework: PyTorch
- Profiler: CodeCarbon

Profiler Selection: CodeCarbon vs Scaphandre

- Compatibility: Scaphandre requires system-level access; incompatible with Kaggle.
- Integration: CodeCarbon easily integrates into Python.
- Cloud-friendly: Works seamlessly with Kaggle GPUs.
- Environmental Metrics: Tracks CO_2 emissions directly.
- Efficiency: Minimal resource overhead compared to Scaphandre

Energy Consumption Analysis





- Carbon Emissions: EfficientNet emits 10x more CO_2 than ResNet-18 due to longer training.
- GPU Power Consumption: ResNet-18 consumes slightly more power per second, but overall emissions are lower.
- EfficientNet took ~2.8 hours to train, while ResNet-18 finished in ~11 minutes, leading to its significantly higher emissions.

Model	Accuracy (20 Epochs)	Energy Consumption
ResNet-18	52.5%	Higher power usage per inference
EfficientNet	99.8%	Optimized computations; lower energy per operation

Profiler Findings on Carbon Footprint

- ResNet-18 emits 0.010 kg CO_2 while EfficientNet emits 0.103 kg CO_2 due to extended training.

Hardware Utilization & Optimization

- Profiler confirms EfficientNet's FLOP-efficient layers reduce per-operation energy use.
- However longer GPU activity increases total energy consumption.
- Batch tuning, quantization, and pruning can cut emissions without sacrificing accuracy.

Strategies for Sustainable AI

- Train during low-carbon energy periods to minimize emissions.
- Implement pruning, quantization, and mixed-precision training for reduced computation.
- Prefer cloud providers with renewable energy sources for AI workloads

EfficientNet

Resnet-18

Software Sustainability & Green AI Principles

Green Software Impact:

Energy Efficiency

- EfficientNet uses optimized computations, reducing power use per operation but requires longer training.
- ResNet-18 trains faster but its design processes more redundant computations, leading to overall lower efficiency.

Resource Optimization

- EfficientNet's design reduces unnecessary computations and power use.
- ResNet-18's older design requires more processing power.

Carbon Sustainability

- EfficientNet lowers GPU usage, reducing CO_2 emissions.
- Using less energy helps cut long-term costs.

Code Sustainability Evaluation – ResNet-18 & EfficientNet

- Clear documentation with explanations and comments. Version control ensures easy tracking and experiment replication.
- Code is organized into separate sections, making it easy to modify and debug.
- CodeCarbon monitors energy use in real time, helping optimize GPU/CPU usage.
- EfficientNet uses advanced techniques to reduce computation and save energy while improving accuracy.
- Code is open-source on GitHub, allowing transparency and contributions from others.

Conclusion

- EfficientNet is more energy-efficient than ResNet-18 while achieving higher accuracy.
- Energy profiling tools help measure and optimize power consumption in AI models.
- Sustainable software practices like modularity and documentation improve long-term usability.
- Adaptive training techniques can further reduce energy waste in AI development.
- Aligning AI workloads with low-carbon energy sources can lower environmental impact.
- Open-source collaboration is key to advancing green AI research and development.

References

- Hugging Face AI Energy Score – Standardized scoring for AI model energy efficiency. (Link)
- Luccioni et al. (2023) – Energy consumption analysis of AI models. (arXiv)
- CodeCarbon Documentation – Tool for tracking AI model CO_2 emissions.
- Green Software Foundation – Articles on sustainable software practices. (Link)
- Software Sustainability Institute (SSI) – Best practices for sustainable software.
- Pereira et al. (2024) – Comparative study of programming languages' energy efficiency. (arXiv)
- Lopes et al. (2017) – Energy efficiency metrics in programming languages. (ACM)
- HasLab SAFER Study – Energy consumption in software execution. (PDF)