Internship Report on

"ANALYSING CARBON FOOTPRINT AND ENERGY EMISSIONS OF LARGE VISION AND AUDIO PROCESSING MODELS"

Submitted to

"CENTER OF COGNITIVE COMPUTING AND COMPUTATIONAL INTELLIGENCE (C3I)"

by

SANJAY BALAJI MAHALINGAM

PES2UG22CS501

Under the guidance of

Dr. Arti Arya

Professor

June 2024 – July 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
Bengaluru – 560 085, Karnataka, India

TABLE OF CONTENTS

INTRODUCTION

The rapid advancements in Artificial Intelligence (AI) and Machine Learning (ML) have led to the development of increasingly complex models, particularly in the fields of Computer Vision and Audio Processing. These models achieve impressive performance but come with a significant environmental cost due to their high computational demands, leading to considerable carbon emissions and global warming.

Given the growing concern over climate change and environmental degradation, it is important to assess the carbon footprint of AI technologies and suggest appropriate mitigation practices. This project aims to address this need by analysing the carbon emissions and energy consumption associated with training and inference of various vision and audio processing models.

Analysing the energy consumption of these models can influence the future development of AI models by providing a comprehensive analysis of the environmental impact of these models and promote the adoption of more energy-efficient practices in AI research and development. Understanding which models are most energy-intensive can guide further research in the fields of Green Computing, Computer Vision and Audio Processing.

PROBLEM STATEMENT

"ANALYSING CARBON FOOTPRINT AND ENERGY EMISSIONS OF LARGE VISION AND AUDIO PROCESSING MODELS"

The primary objective is to analyse the carbon footprint and energy emissions associated with the training and inference of Large Vision and Audio Processing models.

Given the increasing complexity and computational demands of modern AI models, there is a critical need to understand their environmental impact. This study aims to provide a comprehensive analysis of the energy consumption of various models and propose strategies to mitigate their carbon emissions, contributing to the field of sustainable AI development.

ABSTRACT

In this project, the carbon footprint and energy emissions of large vision and audio processing models are analysed. By training and evaluating models like ResNet50, DenseNet121, VGG16, MobileNetV2, EfficientNet, Wave2Vec 2.0, Wav2Letter, DeepSpeech, and HDemucs, we track energy consumption using the CodeCarbon tool. CIFAR-10 dataset is used for vision models and LIBRISPEECH for audio models. Results indicate significant differences in energy usage across models. The report provides insights into optimizing model training for reduced environmental impact and proposes strategies for more sustainable AI practices to reduce global warming.

LITERATURE SURVEY

- Vivian Liu, Yiqiao Yin, “Green AI: Exploring Carbon Footprints, Mitigation Strategies, and Trade Offs in Large Language Model Training”, (2024)
    - paper evaluates carbon emissions of well-known LLMs and highlights importance of hardware in carbon footprint by training and tracking emissions on 2 different GPUs and comparing differences.
    - key info taken from this paper was the usage of **Code Carbon** to track the energy emissions on the GPUs.
    - paper only evaluated the emissions released on **training** specific well known LLMs

- David Patterson, Joseph Gonzalez, Urs Hölzle, Quoc Le, Chen Liang and Lluis-Miquel Munguia, Daniel Rothchild, David R. So, Maud Texier and Jeff Dean, “Carbon Footprint of Machine Learning Training Will Plateau, Then Shrink”, (2022)
    - Paper provides the formula for calculating carbon footprint given energy emissions and carbon intensity.
    - The carbon intensity can vary from region to region and depends on the location of the data centres.

    - Model training and Inference code needs to be used while calculating carbon footprint
    - Energy emissions in this paper are calculated manually by checking the number of hours to train the processor, number of processors and average power per processor.

- Mark A.M. Kramer and Peter M. Roth, “Advancing Green Computer Vision: Principles and Practices for Sustainable Development for Real-time Computer Vision Applications “, (2024)
    - Paper highlights significant issues in deploying computer vision systems in real-world applications such as high computational complexity and power consumption
    - Computational complexity can be reduced by model quantisation (reducing the number of model parameters)
    - Specifies need for accurate and better data collection and preprocessing and prioritizing quality of data over quantity
    - This paper is more for analysis on how to reduce the emissions and carbon footprint

OTHER RESEARCH PAPERS AND ARTICLES REFERRED:

- David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier and Jeff Dean, “Carbon Emissions and Large Neural Network Training”, (2021)

- Emma Strubell, Ananya Ganesh and Andrew McCallum, “Energy and Policy Considerations for Deep Learning in NLP”, (2019)

- Titouan Parcollet, Mirco Ravanelli, “The Energy and Carbon Footprint of Training End-to-End Speech Recognizers”, (2021)

- Joseph McDonald, Baolin Li, Nathan Frey, Devesh Tiwari, Vijay Gadepally, Siddharth Samsi, “Great Power, Great Responsibility: Recommendations for Reducing Energy for Training Language Models”, (2021)

OBJECTIVES:

- Evaluate Carbon Emissions:
    - Measure and compare the carbon emissions and energy consumption of various large vision and audio processing models during training and inference.
    - Assess the efficiency of the training and inference processes for each model, focusing on energy consumption and computational requirements.

- Analyse the Emission of all Models:
    - Identify which models among ResNet50, DenseNet121, VGG16, MobileNetV2, EfficientNet, Wave2Vec 2.0, Wav2Letter, DeepSpeech, and HDemucs exhibit the highest and lowest energy consumption and carbon footprint.
    - Establish relations between energy consumption, number of epochs and time.
    - Perform comparative study on the energy consumption and energy graphs of all models

- Propose Optimization Strategies and Effective Green Computing Strategies:
    - Suggest potential strategies and practices for optimizing the training and inference processes to reduce energy consumption and carbon emissions without compromising model performance.

METHODOLOGY FOLLOWED

Model Selection:

- Vision Models: ResNet50, DenseNet121, VGG16, MobileNetV2, and EfficientNet-B0.

ResNet50**:**

Reason for choosing model:

- ResNet50 is a widely used model in both academic research and industry applications. It serves as a **benchmark** for comparing new architectures due to its well-known performance and robustness.

- Convolution Neural Network consisting of 50 layers used for image classification
- When given an image, ResNet50 processes it through its 50 layers organized into residual blocks.
- The network's residual blocks allow it to learn very deep representations without suffering from vanishing gradients. (uses identity shortcuts)
- the model has around 25.5 million parameters

DenseNet121**:**

Reason for choosing model:

- DenseNet121 is known for its efficient use of parameters through dense connections, which can **potentially** lead to lower energy consumption while maintaining high performance.

- Neural Network consisting of 121 layers.
- Uses dense connections where each layer receives input from all previous layers.
- Has approximately 8 million parameters

VGG16:

Reason for choosing model:

- VGG16 is chosen for its simple and straightforward architecture, which has influenced many subsequent models. Despite its simplicity, it is computationally intensive, making it a good candidate for studying energy consumption.

- deep and uniform architecture with a large number of convolutional layers (16 layers) and fully connected layers.
- Very large number of parameters (approximately 138 million)

- Each convolutional layer has a large number of filters, and the fully connected layers are particularly dense (due to the full connectivity, fully connected layers have a very high number of parameters.)

MobileNetV2

Reason for choosing model:

  - MobileNetV2 is designed for mobile and embedded vision applications, emphasizing low computational cost and high efficiency. Its inclusion allows for studying the trade-offs between model complexity and energy efficiency.

- Uses depthwise separable convolutions and inverted residuals with linear bottlenecks to minimize computational complexity.
- The model outputs class probabilities, indicating the likelihood of the image belonging to each class.
- only 3.4 million parameters (approximately)

EfficientNet-B0

Reason for choosing model:

  - EfficientNet is known for its systematic model scaling approach, which balances network depth, width, and resolution for optimal performance with reduced computational resources.

- Uses a combination of techniques like depthwise separable convolutions and compound scaling to improve efficiency.
- relatively small 5.3 million parameters (approximately)

- Audio Processing Models: Wave2Vec 2.0, Wav2Letter, DeepSpeech, and HDemucs.

Wave2Vec 2.0:

Reason for choosing model:

- Wave2Vec 2.0 is prominent for its self-supervised learning approach, which significantly reduces the need for labeled data, making it an innovative model in speech processing.

- Wave2Vec 2.0 is a model developed by Facebook AI for automatic speech recognition (ASR). It uses a convolutional neural network (CNN) followed by transformer blocks.
- The model learns representations of speech audio through unsupervised pre-training and then fine-tunes on labelled data.

Wav2Letter:

Reason for choosing model:

- Wav2Letter is known for its end-to-end speech recognition capabilities with a relatively simple architecture. Its inclusion helps analyse the efficiency of simpler models compared to more complex ones

- Wav2Letter is a model developed by Facebook AI, known for its simplicity and efficiency. It uses convolutional layers to directly map speech audio to text.
- It's designed to be fast and computationally efficient.

DeepSpeech:

Reason for choosing model:

- DeepSpeech is one of the pioneering models in end-to-end speech recognition, making it essential to study its energy consumption and carbon footprint.

- DeepSpeech is an end-to-end ASR model developed by Mozilla. It uses a combination of convolutional and recurrent neural network (RNN) layers, specifically long short-term memory (LSTM) units, to process audio sequences

HDemucs:

Reason for choosing model:

- HDemucs is designed for high-quality music source separation, a computationally intensive task. Analysing its energy usage provides insights into the cost of complex audio processing tasks.
- HDemucs is designed for audio source separation. This involves isolating different components of an audio signal, such as vocals, drums, bass, and other instruments in a music track.
- It features a hierarchical structure, which means it processes audio in a multi-stage manner, refining the separation progressively at each stage. Typically, it includes encoder-decoder architectures, often with multiple layers and sometimes using mechanisms like attention to improve performance.

Dataset Selection:

- For vision models CIFAR 10 dataset was used.

CIFAR 10:

- CIFAR-10 is a widely used benchmark dataset in the field of Computer Vision. It consists of 60,000 32x32 colour images in 10 classes.
- The dataset is split into 50,000 training images and 10,000 test images.

Reason for choosing CIFAR-10:

- The vision models chosen were pre-trained on ImageNet dataset which is a dataset consisting of 14 million images.
- A smaller dataset had to be chosen for the comparative study but at the same time the dataset also had to be as comprehensive as ImageNet in terms of the variety of images.
- The dataset offers a diverse representation of everyday objects and animals across its 10 classes. This diversity ensures that models trained on CIFAR-10 can generalize to a wide range of real-world visual recognition tasks.

- For audio processing models LIBRISPEECH dataset was used.

LIBRISPEECH

- The LIBRISPEECH dataset is a large-scale corpus of approximately 1,000 hours of 16kHz read English speech. It was introduced in 2015 and has since become a standard benchmark for speech recognition systems.
- The dataset is split into train-clean-100 set which consists of 100 hours of clean training data and a test-clean set which consists of 5.4 hours of testing data.

Reason for choosing LIBRISPEECH:

- Most of the torch audio ASR models were already pre trained on LIBRISPEECH dataset so it was the right dataset to choose for this study as the Audio Processing models we are considering are ASR models
- Not extremely large in size as compared to ImageNet(which the vision models were trained on). So, no replacements needed to be found.

- Dataset Preprocessing:

    - Applied necessary preprocessing steps for each dataset, including normalization, data augmentation, and feature extraction where applicable.

Model Training and Inference:

- Randomized the weights of each model before training. For the vision models, the models had a parameter ‘weights’ so a simple initialization ‘weights=None’ was sufficient. For the audio processing models, the neural networks were traversed layer by layer and the weights were randomized.
- The models were trained on their respective datasets. All vision models were trained on CIFAR 10 dataset with batch size 128 for 100 epochs and all the audio processing models were trained on the LIBRISPEECH dataset with a batch size of 4 for 50 epochs. The batch

sizes were chosen by monitoring GPU memory consumption and ensuring maximum utilization and minimum time consumption.

- The reason that the vision models were trained on 100 epochs but the audio processing models were trained on 50 epochs is because there is a linear relation between the emissions and the number of epochs. This relation is established and justified later in this report. Hence, the emissions of the models on x number of epochs can be extrapolated to the emissions on y number of epochs easily.
- All models were trained on the same hardware (Nvidia GeForce RTX 4090 GPU) to ensure uniformity for comparative study.
- Utilized energy tracking tool CodeCarbon to track the electricity consumption of the GPU at regular intervals during the training process.

CodeCarbon:

- Code Carbon is a python software package that helps calculate the energy consumption while running computationally intensive tasks.
- It tracks the power supply to the underlying hardware at frequent time intervals. This is a configurable parameter with default value 15 seconds, that can be passed when instantiating the emissions' tracker.
- Tracks Nvidia GPUs energy consumption using pynvml library (installed with the package).
- After importing the CodeCarbon library and initializing the emissions tracker, GPU emissions can be tracked using the tracker.start() function.

- The model state is saved after training and the model is loaded again as inference on the test dataset to calculate electricity/energy consumption during inference.

Carbon Footprint Calculation:

- The training and inference energy consumption (in kWh) is added to get total energy consumption. This is done for all the models.
- The energy consumption is multiplied with the carbon intensity (carbon intensity of India is 633.398 kg/MWh) to get the carbon footprint. This carbon intensity of respective countries is mentioned in the Code Carbon documentation and is taken from a reputed source.

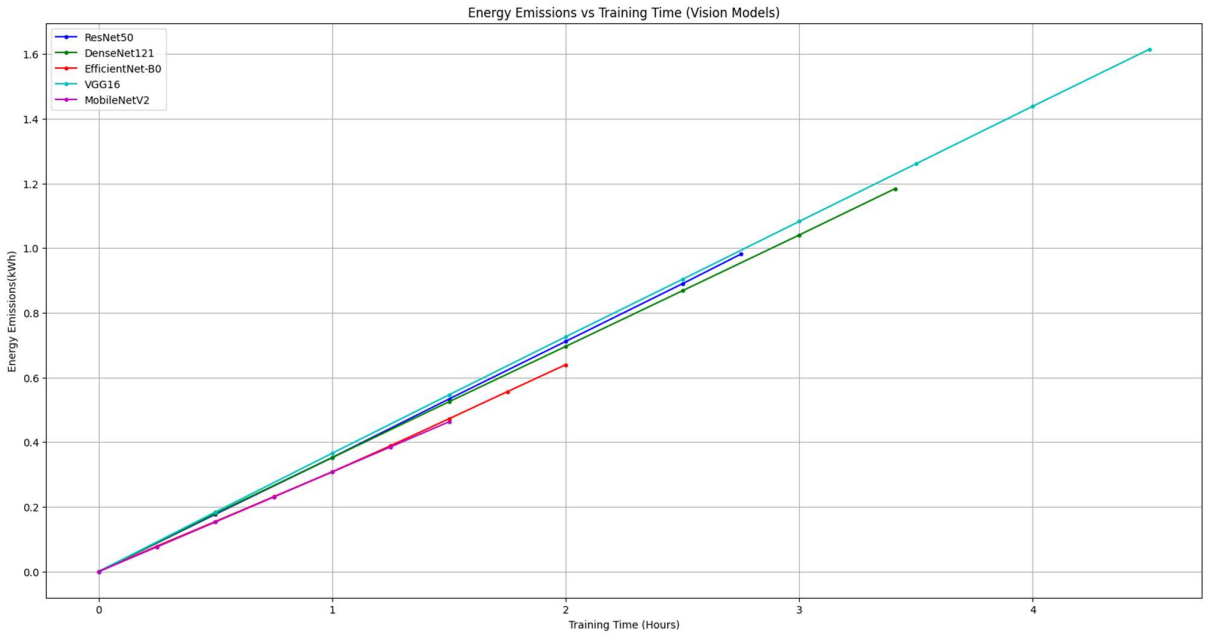
- Carbon footprint of all the models is tabulated to perform comparative study.

Energy Graphs:

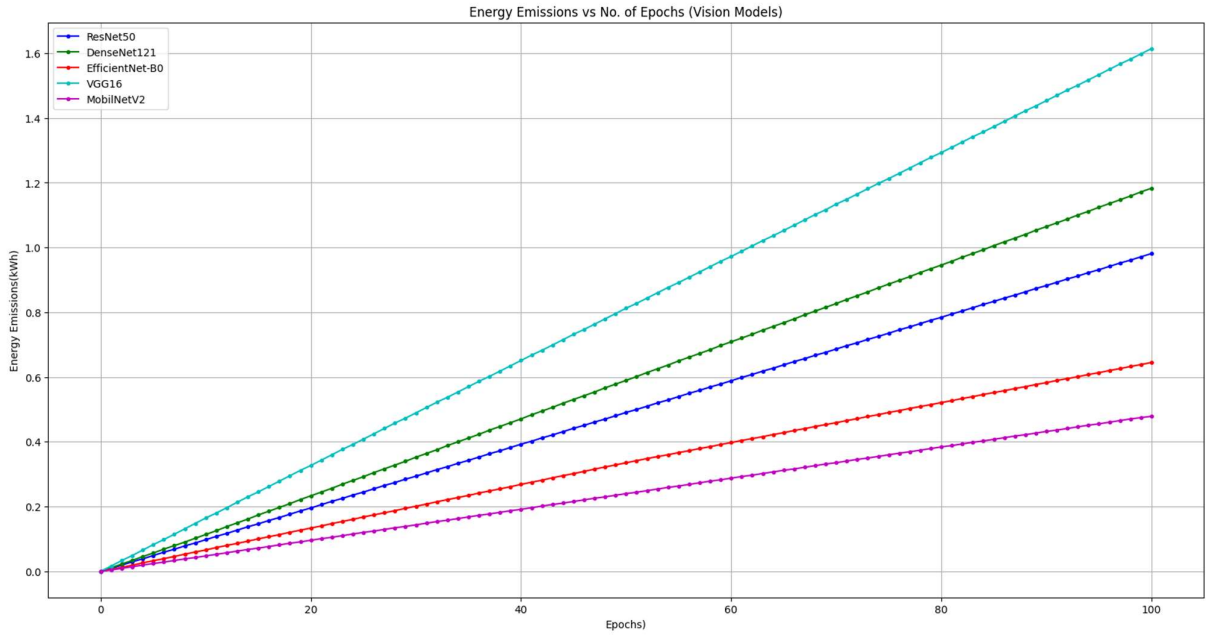
- Energy vs number of epochs and Energy vs Time graphs are constructed for all the models. The data for energy consumption per epoch and energy consumption for regular intervals of time was recorded during the model training.
- Based on this data, inferences can be made on each model's emissions with respect to time and number of epochs.

OUTCOMES OF THE WORK:

The combined energy vs time graph for all vision models is shown below



The combined energy vs epoch graph for vision models is shown below.



OBSERVATIONS (VISION MODELS):

VGG16:

- VGG16 exhibits the highest energy consumption, emitting 1.615 kWh over 100 epochs.
- The Energy-Time graph also shows that it takes the longest time to train (which also corresponds to the linear relation established earlier between energy and time)

Reasons:

- VGG16 has a significant number of parameters (approximately 138 million), which requires more computational resources and thus higher energy.
- The network has three fully connected layers with 4096, 4096, and 1000 neurons, respectively. These layers are dense and involve a high number of operations.
- The model has 13 convolutional layers, making it computationally intensive.

DenseNet121:

- DenseNet121 emits about 1.184 kWh over 100 epochs and takes the second longest time to train. (roughly more than 3 hours)

Reasons:

- DenseNet121 connects each layer to every other layer in a feed-forward fashion, which allows for better gradient flow and potentially more efficient training.
- Despite having dense connections, DenseNet121 has fewer parameters compared to VGG16 (approximately 8 million).
- The lesser number of parameters compared to DenseNet121 and more number of parameters compared to the rest of the models resulted in DenseNet121 falling second on the graph in terms of energy emissions

ResNet50:

- ResNet50 emits 0.981 kWh over 100 epochs with a training time of roughly more than 2 hours

Reasons:

- ResNet50 utilizes residual connections that help in training deeper networks by mitigating the vanishing gradient problem, leading to efficient training. Hence the energy consumption and the time taken to train this model are less
- With around 25.6 million parameters, ResNet50 shows a balance between model complexity and training efficiency.

EfficientNet-B0:

- EfficientNet-B0 emits about 0.645 kWh over 100 epochs with a training time of 2 hours

Reasons:

- EfficientNet-B0 applies a compound scaling method that balances network depth, width, and resolution, leading to efficient use of computational resources.

- With around 5.3 million parameters, EfficientNet-B0 is designed to be efficient while maintaining performance.

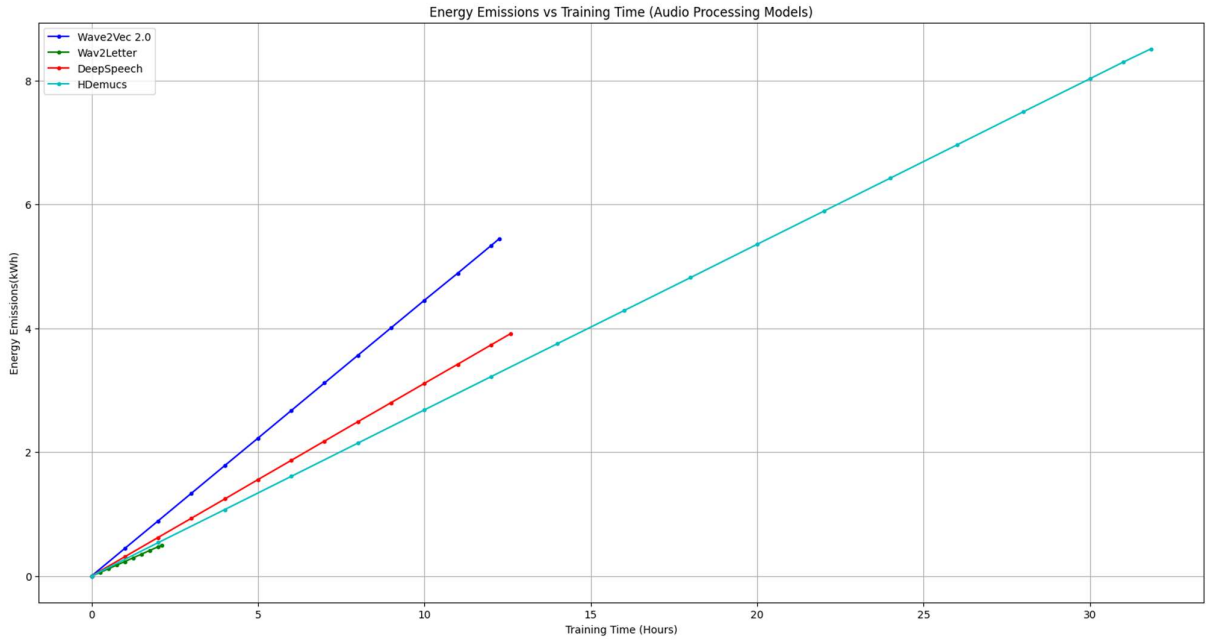
MobileNetV2:

- MobileNetV2 has the lowest energy consumption, emitting approximately 0.479 kWh over 100 epochs taking the shortest time of 1.5 hours.

Reasons:

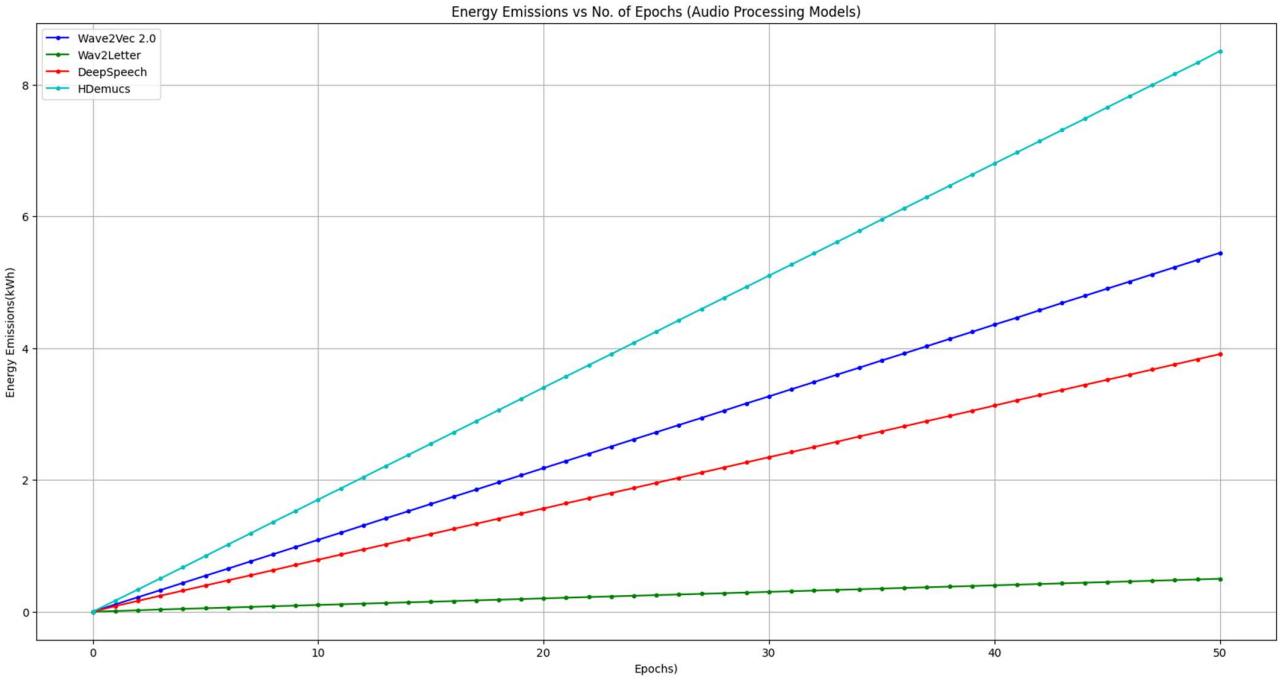
- Depthwise Separable Convolutions: These significantly reduce the number of parameters and computations compared to standard convolutions.
- Inverted Residuals with Linear Bottlenecks: This architecture optimizes the flow of information and reduces the number of operations.
- MobileNetV2 is specifically designed for efficiency, with around 3.4 million parameters.

The combined energy vs time graph for audio processing models is shown below.



The combined energy vs epoch graph for all audio processing models is given below.

.



OBSERVATIONS (AUDIO PROCESSING MODELS):

Hdemucs:

- Energy consumption of 9.988 kWh over 50 epochs with a training time of approximately 31 hours

Reasons:

- The model is designed for music source separation, which is a complex task requiring detailed feature extraction.
- It uses a complex architecture with several layers, leading to high computational requirements.

Wave2Vec 2.0

- Energy consumption 0.499 kWh over 50 epochs with a training time of approximately 12.25 hours

Reasons:

- This model uses convolutional and transformer layers, which are computationally intensive but efficient in processing large datasets.
- The energy consumption is still moderate compared to HDemucs due to the efficiency of the model architecture, but having transformer layers instead of all convolutional layers results in higher computational demand.

DeepSpeech

- Energy consumption over 3.9 kWh over 50 epochs with a training time of approximately 12 hours

Reasons:

- The model uses recurrent neural networks (RNNs), which are computationally demanding due to the sequential nature of processing.
- This results in more computational demand and hence more time needed for training but still better than Wave2Vec and HDemucs.
- Its moderate energy usage indicates a balance between model complexity and computational efficiency.

Wav2Letter:

- Energy consumption over 0.499 kWh over 50 epochs with a training time of approximately 2.25 hours

Reasons:

- It uses a relatively simple and efficient architecture based primarily only on convolutional layers. This design allows it to process audio quickly and with less computational overhead.
- Wav2Letter's design focuses on efficiency, using fewer parameters and simpler computations compared to models that use transformers or RNNs. This results in lower overall energy usage.

BASED ON THE ABOVE STUDY THERE ARE SEVERAL WAYS TO REDUCE CARBON EMISSIONS:

- Model Selection
    - Choose Efficient Architectures: Opt for models like MobileNetV2 and Wav2Letter, which have significantly lower energy consumption. These models are designed to be computationally efficient while maintaining high performance.
- Optimize Training Processes
    - Reducing Epochs: If the model achieves satisfactory performance earlier, consider stopping training before 100 epochs. This can significantly reduce energy consumption

- Efficient Hardware Utilization and Batch Optimization
    - All the models were trained on Nvidia RTX 4090 GPU with a dedicated GPU memory of 24 GB.
    - Different batch sizes were tested to find an optimal balance between training time and computational efficiency.
    - The batch sizes chosen while training the vision models was 128 and the batch sizes chosen for training the audio processing models was 4. This was done observing the GPU memory and ensuring maximum possible GPU utilization.
    - However more efficient hardware always exist and should be incorporated to reduce carbon emissions.
- Model Pruning and Quantization
    - Model Pruning: Remove unnecessary weights and neurons from the network to reduce its size and complexity, leading to lower energy consumption during training and inference.

- Quantization: Convert the model weights from floating-point precision to lower precision (e.g., int8), which reduces computational load and energy usage.

- Transfer Learning:
  - Utilize pre-trained models and fine-tune them on specific tasks, which requires significantly less energy compared to training from scratch.
- Green Data Centres:
  - Train models in data centres powered by renewable energy sources. Many cloud providers now offer options to choose data centres with greener energy profiles.

CONCLUSION:

- Significant Variability in Energy Consumption and Carbon Emissions:
  - This study revealed considerable differences in energy consumption and carbon emissions across different vision and audio processing models. VGG16 and HDemucs exhibited the highest emissions, highlighting the impact of model complexity on energy usage.
- Efficiency of Models:
  - MobileNetV2 and Wav2Letter, which are designed with efficiency in mind, demonstrated lower energy consumption compared to traditional models like VGG16 and HDemucs. This underscores the importance of model architecture in achieving energy efficiency.
- Optimization Strategies:
  - Selecting the right models, implementing optimization techniques such as model pruning, quantization, and efficient training algorithms can substantially reduce the carbon footprint of AI models without compromising performance. These strategies should be prioritized in future model development.
- Broader Implications:

- This study contributes to the growing body of research on Green AI, highlighting the importance of reducing the environmental impact of AI technologies.

REFERENCES:

- Vivian Liu, Yiqiao Yin, “Green AI: Exploring Carbon Footprints, Mitigation Strategies, and Trade Offs in Large Language Model Training”, (2024)
- David Patterson, Joseph Gonzalez, Urs Hölzle, Quoc Le, Chen Liang and Lluis-Miquel Munguia, Daniel Rothchild, David R. So, Maud Texier and Jeff Dean, “Carbon Footprint of Machine Learning Training Will Plateau, Then Shrink”, (2022)

- Mark A.M. Kramer and Peter M. Roth, “Advancing Green Computer Vision: Principles and Practices for Sustainable Development for Real-time Computer Vision Applications “, (2024)

- CodeCarbon

  (https://codecarbon.io/)

- Vision Models
  (https://pytorch.org/vision/0.9/models.html)

- Audio Models
  (https://pytorch.org/audio/stable/models.html)

- CIFAR 10 dataset
  (https://pytorch.org/vision/stable/generated/torchvision.datasets.CIFAR10.html#torchvision.datasets.CIFAR10)

- LIBRISPEECH dataset
  (https://pytorch.org/audio/stable/generated/torchaudio.datasets.LIBRISPEECH.html#torchaudio.datasets.LIBRISPEECH)