11TH INTERNATIONAL SCIENTIFIC CONFERENCE
ON DEFENSIVE TECHNOLOGIES
OTEH 2024
Serbia, Tara, 9-11 October 2024

THE POWER CONSUMPTION OF EMBEDDED GPU COMPUTERS FOR DEEP LEARNING APPLICATIONS

BOBAN SAZDIĆ-JOTIĆ
Military Technical Institute, Belgrade, boban.sazdic.jotic@vs.rs

SNEŽANA ZUROVAC
Military Technical Institute, Belgrade, zurovac@medianis.net

NIKOLA PETROVIĆ
Military Technical Institute, Belgrade, npetrovicbrg@gmail.com

DRAGANA BOJIĆ
Military Technical Institute, Belgrade, dragana.bojic@mod.gov.rs

Abstract: *The widespread use of GPU processors has significantly enhanced the usage of various artificial intelligence algorithms, especially machine and deep learning models. However, more capabilities introduced significant challenges in energy consumption, particularly on autonomous platforms with battery and autonomy constraints. Our research is crucial as it addresses the energy consumption issue on desktop and embedded GPU computers from a new angle, providing a practical and optimal measurement solution. We engaged various deep learning models to analyze and determine how different datasets for training and problems for inference affect energy consumption. Two separate platforms were used for the experiments: the NVIDIA Jetson NANO platform, a well-known type of embedded GPU computer widely used in deep learning applications, for the inference process just, and a desktop computer with two GeForce RTX 2060 for the training and inference processes. The image classification problem was involved during the training process on two separate datasets. In contrast, inference problems for object detection and drone classification were engaged on live videos and recorded RF signals during the inference process. Our methodology involved a systematic comparison of energy consumption across different models and datasets, ensuring the validity and reliability of our findings. Our findings underscore the potential to reduce energy consumption on embedded GPU computers by implementing a suitable deep learning model. This not only preserves the performance of the required process (drone radio frequency signal detection and identification and object detection in the video stream) but also paves the way for their effective use in real-life scenarios, thereby addressing a crucial need in deep learning. More importantly, we created an empirical methodology for measuring the energy consumption of embedded GPU computers for deep learning applications, directly impacting the development of energy-efficient deep learning systems.*

Keywords: *artificial intelligence, deep learning, drone, detection, embedded GPU computer, energy consumption.*

1. INTRODUCTION

Artificial intelligence (AI) algorithms are used in various fields, such as industry, medicine, agriculture, and defense, to automate processes and facilitate decision-making procedures. A prerequisite for such rapid development is, among other things, the development of computer science, primarily the upgrading of Graphics Processing Unit (GPU) processors and improvements in effective parallel processing of substantial amounts of data. Deep Neural Networks (DNN) have enabled state-of-the-art accuracy on many challenging artificial intelligence tasks, such as image classification and segmentation [1], [2], [3], object detection and tracking [4], [5], [6], [7], natural language processing [8], [9], [10], [11], [12], [13], and voice-based applications [14], [15].

However, this advanced technology is highly dependent on high-performance computing (HPC) based on GPU processors and has an evident downside: it consumes significant energy. On the other hand, the authors in [16] have suggested putting a goal on the environmentally friendly deep learning (DL) models known as Green AI. To accomplish that, a huge problem arises when there is a demand to apply DL models to mobile applications where energy consumption is crucial for implementation [17], [18]. While most AI computation for training purposes currently resides online (cloud or local servers) or on desktop computers, using embedded GPU computers for inference locally near the sensor is desirable due to privacy, security, and latency concerns or limitations in communication bandwidth. Since mobile devices require batteries, reducing power consumption alleviates environmental problems and extends battery life, making low-power DL algorithms highly desirable for different use cases. In addition, a survey presented in [19] reveals that approximately 55% of respondents would leave a

Corresponding author: B.Sazdić Jotić;
E-mail: boban.sazdic.jotic@vs.rs
DOI: 10.5937/OTEH24069S

negative review for a mobile application that drains a significant amount of battery. A similar prerequisite exists in functional applications, especially for military purposes where embedded and mobile systems must respond accordingly to responsive energy consumption. For example, military drones have limited batteries for flying operations and DL model inference. Accordingly, the research community has been increasingly interested in designing energy-efficient DL models, critical to realizing mobile AI applications. However, estimating energy consumption from the DL model is much more complicated than other metrics, such as storage cost (model size), throughput (number of operations), or latency (time of processing). This is because a huge portion of the energy is consumed by data movement, which is difficult to extract directly from the DL model.

Because of this, our research presents a study of energy consumption for DL models, especially methods that can be used for initial estimation. This paper explores how energy consumption is affected by using different GPU processors and DL models in two distinct stages of production (training or inference). This paper is structured as follows: Section 2 provides an overview of the methods for energy estimation. Sections 3 and 4 present the methodology and discuss our experimental results, and the study conclusion is given in the last section.

2. CURRENT ENERGY ESTIMATION METHODS

Energy consumption in embedded GPU computers is a prominent concern in computational research, and multiple studies have explored various optimization strategies. Based on our comprehension, current energy estimation methods can be classified into three groups.

The first group of papers incorporates an approach to reducing consumption by knowing computer operation processes and the lowest levels of computer computation to optimize calculations performed on the computer platform, primarily on the GPU processor. This group includes papers that predict the number of computational and memory access operations to predict energy consumption. The new energy monitoring and optimization capabilities as an autotuning tool for GPU processors were presented in [20]. Another approach called the SyNERGY was introduced in [21], which uses an energy measurement based on hardware performance counters such as SIMD (single instruction, multiple data) and bus access for the CPU obtained from actual execution runs of these models. The authors in [22] have proposed a methodology that can be used to evaluate the various DL models and energy-efficient techniques, with a guide for designing the energy-efficient DL models. They pointed out that the DL model's convolutional and fully connected layers dominated computation and energy consumption. An interesting method is introduced in [23] with the idea that it cannot optimize what cannot be measured and cannot optimize what is under-appreciated or neglected in the design. This is even important because most research tends to reduce inference latency and enhance accuracy, often failing to consider the impact on energy efficiency.

The second group of papers includes empirical methods focusing on the results obtained through various experiments or comparing theoretical considerations and practical measurements. Authors in [24] presented measurements of the energy consumption on the Tier-2 HoreKa supercomputing system, presenting that model training and inference energy consumption should be considered separately. A comprehensive empirical method is given in [25] to profile the energetic consumption of inference tasks for some modern edge computers such as Asus Tinker Edge R [26], Raspberry Pi 4 [27], Google Coral Dev Board [28], Nvidia Jetson Nano [29], and one microcontroller Arduino Nano 33 BLE [30] with different DL models. The impact of the DNN architecture on the energy consumption and emissions produced, the trade-off between accuracy and energy efficiency, and the difference in the measurement method of the energy consumed using software-based and hardware-based tools were investigated in [31].

The third group of papers has focused on the development of separate AI models for predicting the level of energy consumption. It should be noted that this approach has also been used for other applications, such as prediction and reduction of energy consumption by the data centers. The authors in [32] have presented a practical framework based on the grid search to answer the above question. Additionally, authors in [33] have created BUTTER and BUTTER-E datasets to characterize the impact of hyperparameter choice on energy efficiency. Their research concludes that smaller networks sometimes consume less energy during training, and broad layers, particularly those with large input sizes, can be power consumers. Moreover, there are more benchmark datasets, such as NAS-Bench 101 [34]. This dataset evaluates various DL models and their associated loss and accuracy scores after training. Also, the authors in [35] estimated energy consumption and carbon emissions of DL models with the EC-NAS benchmark.

3. EXPERIMENTS

The methodology used in this study was designed to perform the following: hardware and software configuration (data and model loading), training or inference process, and analysis of the energy consumption results. Nine different models (AlexNet, InceptionV3, VGG16, VGG19, ResNet50, three various versions of YOLO models, and VTI_CNN – our custom-made convolutional neural network) were initially configured with three different datasets (MINST [36], CIFAR-10 [37], and VTI_DroneSET [38]). We measured CO2 (carbon dioxide) total emission, energy consumption, FLOPS (floating-point operations per second), and accuracy for each model during training and inference processes.

Hardware. All measurements within this research were conducted on an embedded GPU computer, the Jetson Nano Developer Kit, and a desktop GPU computer with

two GeForce RTX 2060. NVIDIA leverages its GPU processors in the Jetson product line to deliver accelerated AI performance to edge platforms. The Jetson Nano's embedded GPU computer is developed based on the Maxwell architecture and incorporates a single streaming processor with 128 CUDA (compute unified device architecture) cores. Positioned as the lowest-end model in the product line, the Jetson Nano achieves a peak performance of 472 Gigaflops. [29]. We reviewed the 4GB model, while there are 2GB and 4GB RAM options with 5W and 10W power modes, respectively. The GeForce RTX 2060 is based on the Turing architecture GPU processor with 6 GB of GDDR6 RAM running at 14 Gbps and a GPU clock of 1.68 GHz. Moreover, it features 1,920 CUDA cores and 240 Tensor Cores that can deliver 52 Teraflops of DL horsepower. The GeForce RTX 2060 has a total board power of 160W.

According to the authors in [39] the most effective method to measure accurate and fine-grained power consumption is to connect the device to an external power monitor (wattmeter) with a high sampling rate. Moreover, the same authors have set rules for measuring power to ensure consistency and reliability across different devices and testing conditions. Controlling environmental factors helps better understand power and energy consumption in DNN executions, so similar rules have been adopted for measurements:

- All wireless communication interfaces (Wi-Fi, Bluetooth, cellular network, and Near-Field Communication) and background applications and services were shut down to minimize interference with measurement accuracy.
- The Jetson Nano Developer Kit was used in headless mode (without monitor and peripherals) due to the more realistic conditions.
- Room temperature was set to be between 20 and 25°C, while the cooldown interval between individual experiments was set to be 5 minutes.

The external power meters used for the experiment are the Fluke 430-II Power Quality and Energy Analyzer and the Acuvim 3 Power Quality and Revenue Grade Energy Meter. However, due to the complicated setup and mandatory hardware adjustment, the experiments also used the CodeCarbon energy profiler [31], which is representative of internal power monitors (energy profilers). The study compares and cross-references their readings with those from an external.

Software. Two requests were considered within the experiments: Does the convolutional neural network (CNN) architecture impact energy consumption, and can general demands for embedded GPU computers be pointed out? We ground our experiments on publicly available models such as AlexNet [40], VGG [41], InceptionV3 (GoogLeNet) [42], ResNet50 [43] and YOLO models [44]. These models were selected as representative CNN models due to their various applications. We also evaluated our custom-made CNN model, specially designed for specific tasks (drone classification in the frequency domain).

In CNN models, the convolutional layer is crucial for extracting and refining features from input data. The energy consumption disparity between external memory access and computational (MAC) operations has led to integrating multiple levels of local memory within GPU processors, forming a memory hierarchy. This hierarchy reduces energy costs associated with external memory access but requires more energy than computation. As a result, the volume of weight movement may provide a more precise estimate of energy consumption. Therefore, current research in efficient CNN design primarily concentrates on creating empirical metrics for energy-efficient CNN models and leads to possibilities to reduce the number of filter weights and the number of multiplication and accumulation operations. It is important to note that these metrics do not necessarily correspond to energy consumption because energy consumption is influenced by data movement rather than computation, and the energy consumption of data movement is significantly impacted by the memory hierarchy and data flow.

The CNN models mentioned above were used to solve the two problems based on such prerequisites. The first is classifying objects on the video, and the second is classifying radio signals in the frequency domain. Even if we define such problems, it can be generalized that all results can be applied to the computer vision domain. All models and test procedures were built in Python. We used the CodeCarbon profiler, a Python package that enables us to track four numerical variables: 1) the emissions in carbon dioxide equivalent (CO2-eq) in kilograms, 2) the energy consumed by the infrastructure in kilowatt-hours, 3) the FLOPS required to train the model, and 4) the validation accuracy of the model obtained. The CodeCarbon energy profiler is based on software metrics and estimates the energy consumed during a model's training to estimate actual energy consumption. reasonably [31].

The CO2-eq is a measure used to compare emissions from different greenhouse gases based on their global warming potential [45]. This value is calculated by multiplying the carbon intensity of the electricity used for computation (measured in kg of CO2 emitted per kWh of electricity) by the net power supply consumed by the computational infrastructure (measured in kWh). Monitoring the power supply to the hardware occurs regularly and depends heavily on the type of hardware used.

Energy consumption is intricately linked to CO2 emissions and is invariant concerning time and location. Monitoring the power supply to the hardware transpires frequently and is heavily contingent upon the type of hardware employed. The experiment was repeated three times, and the median energy consumption value, determined by both the wattmeter and the profiler, is being reported.

FLOPS epitomizes the number of floating-point operations per second necessary to execute a computational process. They estimate the process's workload based on a deterministic measure calculated by tabulating the costs of two fundamental operations:

addition and multiplication. FLOPS can be calculated using a model instance even before the commencement of training.

The overall CNN model score is finally calculated with the following equation:

$$CNN_{score} = \frac{CNN_{accuracy}}{Energy_{consumption}} \quad (1)$$

where $CNN_{accuracy}$ is the accuracy of the used CNN model and $Energy_{consumption}$ is measured energy consumption. The results from the training and inference process and external and profilers' energy measurements were analyzed concurrently to make a definitive conclusion.

4. EXPERIMENTAL RESULTS

Every experiment (training or inference) was performed in three independent executions (K=3) due to the statistical purposes and situation of hardware malfunctioning. Table 1 shows the average emissions produced.

Table 1. The average CO2-eq emissions.

CNN model	CO2-eq emissions [kg emitted per kWh of electricity]	
	CFAR10 dataset [32x32 / 60.000 images]	MNIST dataset [28x28 / 60.000 images]
AlexNet	0.000039373	0.000043659
VGG16	0.006836938	0.000047602
VGG19	0.007825523	0.000049091
ResNet50	0.001236008	0.000056854
InceptionV3	0.000061455	0.000074992

The VGG16, VGG19, and ResNet50 models have the worst results for used hardware configuration due to the highest CO2-eq emissions (CO2-eq or CO2 equivalent) expressed in kg emitted per kWh of electricity. However, the CO2-eq emission also depends on the dataset used for the training process. This is an expected outcome since external memory access in the case of the CFAR10 dataset requires more energy than the MNIST dataset. Interestingly, the CO2-eq emission results can fluctuate significantly for the CFAR10 dataset. The overfitting of these models can explain this phenomenon because they tend to learn details that are not important in the classification process. Moreover, this can be supported by the achieved accuracy of these models for CIFAR (see Table 2).

Tables 2 to 4 show the accuracy obtained from different CNN models, the two distinct stages of production (training or inference), the energy consumption measurements for internal or external power monitors, and the score corresponding to the ratio between the mentioned accuracy and power.

Table 2. Scores of the CNN models for training process on the desktop with two GeForce RTX 2060.

	Model	FLOPS [Gigaflops]	Datasets	Accuracy [%]	Internal power monitor		External power monitor	
					Energy consumption [kWh]	Score	Energy consumption [kWh]	Score
Training process [K=3] for image classification problem	AlexNet	39.1	*CFAR10*	*75.16*	*0.01527*	*49.28333*	*0.014*	*35.12762*
			MNIST	96.71	0.01706	56.89079	0.016	60.61176
	VGG16	21.3	*CFAR10*	*60.03*	*0.01090*	*55.52622*	*0.042*	*50.38429*
			MNIST	93.55	0.01863	43.24232	0.047	59.95124
	VGG19	26.7	*CFAR10*	*59.04*	*0.01256*	*47.11753*	*0.015*	*39.46621*
			MNIST	93.23	0.01942	48.15551	0.016	57.42648
	ResNet50	6,6	*CFAR10*	*20.40*	*0.01359*	*15.11031*	*0.015*	*13.31681*
			MNIST	87.63	0.02235	39.32731	0.016	53.71520
	InceptionV3	14.5	*CFAR10*	*65.01*	*0.02432*	*26.79580*	*0.021*	*30.61719*
			MNIST	89.20	0.03052	29.27427	0.018	48.68782

Table 3. Scores of the CNN models for inference processes on the Jetson Nano Developer Kit for the object detection problem.

	Model	FLOPS [Gigaflops]	Problem	Model confidence [%]	Internal power monitor		External power monitor	
					Energy consumption [kWh]	Score	Energy consumption [kWh]	Score
Inference process [K=3]	YOLOv5s	16.5	Object detection on live video	95.00	0.09704	9.78977	0.085	3.72752
	YOLOv7	36.9			0.13377	7.10175	0.149	6.37583
	YOLOv8s	28.6			0.03511	27.05781	0.031	30.64516

Table 4. Scores of VTI_CNN model for the drone classification problem on the different GPU processors.

	GPU Computer	CNN model	FLOPS [Gigaflops]	Problem	Accuracy [%]	Internal power monitor		External power monitor	
						Energy consumption [kWh]	Score	Energy consumption [kWh]	Score
Inference process [K=3]	2 x GeForce RTX 2060	VTI_CNN	1.7	Drone classification on recorded RF signals	98.18	0.04697	21.04055	0.05500	18.40278
	Jetson Nano Developer Kit				99.57	0.13066	7.62015	0.11588	8.59185

Table 2 shows the average scores of the CNN models for the three (K=3) training processes on the desktop with two GeForce RTX 2060 for solving classification problems with two datasets containing ten classes. The best result was obtained with the AlexNet model for the MNIST dataset and the VGG16 for the CIFAR-10 dataset. These results are consistent with findings from the literature, demonstrating that the proposed methodology can be utilized for further analysis. In [31], the authors compared three models and suggested that the VGG16 model outperformed the VGG19 and ResNet50.

Our research involved five models within the CIFAR-10 and MNIST datasets and indicated the same results, with the addition that the AlexNet model achieved the highest score compared with the models mentioned. Moreover, the results from internal and external power monitors are similar, which leads to the conclusion that internal monitors can be successfully used for general purposes because of their simplicity.

Table 3 shows the average scores of the YOLO models for the three (K=3) training processes on the Jetson Nano Developer Kit for the object detection problem in the inference process within live video. The model's confidence in its prediction accuracy (indicating how confident it is that the predicted results correspond to its labeled class) for all measurements was 95%. The best results are obtained for the YOLO 8 model. Table 4 shows the average scores of the VTI_CNN model for the drone classification problem on recorded RF signals. Better results are obtained with the desktop configuration because of the required external MAC operations necessary for the inference process. MAC operations in the Jetson Nano Developer Kit are more demanding because a dedicated pipeline for images streaming from the external memory to the model is needed during the inference process.

5. CONCLUSION

This study presents an empirical energy estimation methodology for CNN based on the architecture, dataset (MINST, CIFAR-10, and VTI_DroneSET), and production problem (object detection on live videos or drone classification on recorded RF signals). We observed varying overall CNN model scores for the same dataset, demonstrating that the CNN architecture influences energy consumption. AlexNet and VGG16 yielded the best performance during training, while YOLO 8 excelled during inference. As a general observation, we established that internal energy profilers can effectively estimate energy consumption, offering crucial initial insights, especially in military mobile applications. Using this methodology and tool, researchers can quantify the energy consumption on the embedded GPU computers associated with various model choices during the initial design phase. This narrows the gap between CNN algorithm/hardware design and energy optimization. Further research lines can focus on experiments in different temperature conditions to find the relationship with this variable. Most embedded GPUs are intended for external applications, so detecting this impact is essential.

References

[1] H. Sun, X. Zheng, and X. Lu, "A Supervised Segmentation Network for Hyperspectral Image Classification," *IEEE Transactions on Image Processing*, vol. 30, pp. 2810–2825, 2021, doi: 10.1109/TIP.2021.3055613.

[2] Y. Xiao, L. Daniel, and M. Gashinova, "Image Segmentation and Region Classification in Automotive High-Resolution Radar Imagery," *IEEE Sens J*, vol. 21, no. 5, pp. 6698–6711, Mar. 2021, doi: 10.1109/JSEN.2020.3043586.

[3] J. Cheng *et al.*, "ResGANet: Residual group attention network for medical image classification and segmentation," *Med Image Anal*, vol. 76, p. 102313, Feb. 2022, doi: 10.1016/j.media.2021.102313.

[4] T. Yin, X. Zhou, and P. Krähenbühl, "Center-based 3D Object Detection and Tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11779–11788. doi: 10.1109/CVPR46437.2021.01161.

[5] R. Ravindran, M. J. Santora, and M. M. Jamali, "Multi-Object Detection and Tracking, Based on DNN, for Autonomous Vehicles: A Review," Mar. 01, 2021, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/JSEN.2020.3041615.

[6] L. Wen *et al.*, "UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking," *Computer Vision and Image Understanding*, vol. 193, p. 102907, Apr. 2020, doi: 10.1016/j.cviu.2020.102907.

[7] Y. Wang, V. Ilic, J. Li, B. Kisačanin, and V. Pavlovic, "ALWOD: Active Learning for Weakly-Supervised Object Detection," in *Proceedings of the IEEE International Conference on Computer Vision*, IEEE, Oct. 2023, pp. 6436–6446. doi: 10.1109/ICCV51070.2023.00594.

[8] I. Popovic, D. Culibrk, M. Mirkovic, and S. Vukmirovic, "Automatic Speech Recognition and Natural Language Understanding for Emotion Detection in Multi-party Conversations," in *MuCAI 2020 - Proceedings of the 1st International Workshop on Multimodal Conversational AI*, Association for Computing Machinery, Inc, Oct. 2020, pp. 31–38. doi: 10.1145/3423325.3423737.

[9] A. Galassi, M. Lippi, and P. Torroni, "Attention in Natural Language Processing," *IEEE Trans Neural Netw Learn Syst*, vol. 32, no. 10, pp. 4291–4308, Oct. 2021, doi: 10.1109/TNNLS.2020.3019893.

[10] Y. Kang, Z. Cai, C. W. Tan, Q. Huang, and H. Liu, "Natural language processing (NLP) in management research: A literature review," Apr. 02, 2020, *Taylor & Francis*. doi: 10.1080/23270012.2020.1756939.

[11] G. G. Chowdhury, "Natural language processing," *Annual Review of Information Science and Technology*, vol. 37, pp. 51–89, 2003, doi: 10.1002/aris.1440370103.

[12] M. Zhou, N. Duan, S. Liu, and H. Y. Shum, "Progress in Neural NLP: Modeling, Learning, and Reasoning," Mar. 01, 2020, *Elsevier*. doi: 10.1016/j.eng.2019.12.014.

[13] D. Ofer, N. Brandes, and M. Linial, "The language of proteins: NLP, machine learning & protein sequences," Jan. 01, 2021, *Elsevier*. doi: 10.1016/j.csbj.2021.03.022.

[14] K. Chachadi and S. R. Nirmala, "Voice-Based Gender Recognition Using Neural Network," in *Lecture Notes in Networks and Systems*, vol. 191, Springer, Singapore, 2022, pp. 741–749. doi: 10.1007/978-981-16-0739-4_70.

[15] A. Ouhmida, O. Terrada, A. Raihani, B. Cherradi, and S. Hamida, "Voice-Based Deep Learning Medical Diagnosis System for Parkinson's Disease Prediction," in *2021 International Congress of Advanced Technology and Engineering, ICOTEN 2021*, Institute of Electrical and Electronics Engineers Inc., Jul. 2021. doi: 10.1109/ICOTEN52080.2021.9493456.

[16] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green AI," *Commun ACM*, vol. 63, no. 12, pp. 54–63, Nov. 2020, doi: 10.1145/3381831.

[17] H. Wang, B. G. Kim, J. Xie, and Z. Han, "LEAF + AIO: Edge-Assisted Energy-Aware Object Detection for Mobile Augmented Reality," *IEEE Trans Mob Comput*, vol. 22, no. 10, pp. 5933–5948, Oct. 2023, doi: 10.1109/TMC.2022.3179943.

[18] H. Wang and J. Xie, "User Preference Based Energy-Aware Mobile AR System with Edge Computing,"

Proceedings - IEEE INFOCOM, vol. 2020-July, pp. 1379–1388, Jul. 2020, doi: 10.1109/INFOCOM41043.2020.9155517.

[19] Verizon Media, "Apigee Survey: Users Reveal Top Frustrations That Lead to Bad Mobile App Reviews." Accessed: Jul. 15, 2024. [Online]. Available: https://sg.finance.yahoo.com/news/apigee-survey-users-reveal-top

[20] R. Schoonhoven, B. Veenboer, B. Van Werkhoven, and K. J. Batenburg, "Going green: optimizing GPUs for energy efficiency through model-steered auto-tuning," *Proceedings of PMBS 2022: Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems, Held in conjunction with SC 2022: The International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 48–59, 2022, doi: 10.1109/PMBS56514.2022.00010.

[21] C. Faviola Rodrigues, G. Riley, and M. Luján, "SyNERGY: An energy measurement and prediction framework for Convolutional Neural Networks on Jetson TX1," *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA)*, pp. 375–382, 2018.

[22] T. J. Yang, Y. H. Chen, J. Emer, and V. Sze, "A method to estimate the energy consumption of deep neural networks," *Conference Record of 51st Asilomar Conference on Signals, Systems and Computers, ACSSC 2017*, vol. 2017-Octob, pp. 1916–1920, 2017, doi: 10.1109/ACSSC.2017.8335698.

[23] X. Tu *et al.*, "Unveiling Energy Efficiency in Deep Learning: Measurement, Prediction, and Scoring Across Edge Devices," *Proceedings - 2023 IEEE/ACM Symposium on Edge Computing, SEC 2023*, pp. 80–93, 2023, doi: 10.1145/3583740.3628442.

[24] R. Caspart *et al.*, "Precise Energy Consumption Measurements of Heterogeneous Artificial Intelligence Workloads," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13387 LNCS, pp. 108–121, 2022, doi: 10.1007/978-3-031-23220-6_8.

[25] S. P. Baller, A. Jindal, M. Chadha, and M. Gerndt, "DeepEdgeBench: Benchmarking Deep Neural Networks on Edge Devices," *Proceedings - 2021 IEEE International Conference on Cloud Engineering, IC2E 2021*, pp. 20–30, 2021, doi: 10.1109/IC2E52221.2021.00016.

[26] ASUS, "Tinker Edge R," ASUS. Accessed: Jul. 15, 2024. [Online]. Available: https://tinker-board.asus.com/series/tinker-edge-r.html

[27] Raspberry, "Raspberry Pi 4 Model B," Raspberry Pi. Accessed: Jul. 15, 2024. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/

[28] Google, "Dev Board Coral," Google. Accessed: Jul. 15, 2024. [Online]. Available: https://coral.ai/products/dev-board/

[29] NVIDIA, "Jetson Nano," NVIDIA. Accessed: Jul. 15, 2024. [Online]. Available: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/product-development/

[30] Arduino, "Arduino Nano 33 BLE," 2020. Accessed: Jul. 15, 2024. [Online]. Available: https://store.arduino.cc/products/arduino-nano-33-ble

[31] Y. Xu, S. Martínez-Fernández, M. Martinez, and X. Franch, "Energy Efficiency of Training Neural Network Architectures: An Empirical Study," *Proceedings of the Annual Hawaii International Conference on System Sciences*, vol. 2023-Janua, pp. 781–790, Feb. 2023, [Online]. Available: http://arxiv.org/abs/2302.00967

[32] M. Yan, H. Wang, and S. Venkataraman, "PolyThrottle: Energy-efficient Neural Network Inference on Edge Devices," *ArXiv*, vol. abs/2310.1, Oct. 2023, [Online]. Available: https://api.semanticscholar.org/CorpusID:264824281

[33] C. E. Tripp *et al.*, "Measuring the Energy Consumption and Efficiency of Deep Neural Networks: An Empirical Analysis and Design Recommendations," *ArXiv*, vol. abs/2403.0, pp. 1–25, Mar. 2024, [Online]. Available: http://arxiv.org/abs/2403.08151

[34] C. Ying, A. Klein, E. Real, E. Christiansen, K. Murphy, and F. Hutter, "NAS-Bench-101: Towards Reproducible Neural Architecture Search," *36th International Conference on Machine Learning, ICML 2019*, vol. 2019-June, pp. 12334–12348, Feb. 2019, Accessed: Jul. 15, 2024. [Online]. Available: https://proceedings.mlr.press/v97/ying19a.html

[35] P. Bakhtiarifard, C. Igel, and R. Selvan, "EC-NAS: Energy Consumption Aware Tabular Benchmarks for Neural Architecture Search," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, Apr. 2024, pp. 5660–5664. doi: 10.1109/ICASSP48485.2024.10448303.

[36] L. Deng, "The MNIST database of handwritten digit images for machine learning research," *IEEE Signal Process Mag*, vol. 29, no. 6, pp. 141–142, Nov. 2012, doi: 10.1109/MSP.2012.2211477.

[37] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009. Accessed: Aug. 30, 2024. [Online]. Available: http://www.cs.utoronto.ca/~kriz/learning-features-2009-TR.pdf

[38] B. M. Sazdić-Jotić *et al.*, "VTI_DroneSET," Mendeley Data. Accessed: Nov. 01, 2020. [Online]. Available: https://data.mendeley.com/datasets/s6tgnnp5n2/1

[39] A. Pathak, Y. C. Hu, and M. Zhang, "Where is the energy spent inside my app?: Fine grained energy accounting on smartphones with eprof," *EuroSys'12 - Proceedings of the EuroSys 2012 Conference*, pp. 29–42, 2012, doi: 10.1145/2168836.2168841.

[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun ACM*, vol. 60, no. 6, pp. 84–90, 2017, doi: 10.1145/3065386.

[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations*, International Conference on Learning Representations, ICLR, 2014, pp. 1–14. doi: 10.48550/arxiv.1409.1556.

[42] C. Szegedy *et al.*, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2015, pp. 1–9. doi: 10.1109/CVPR.2015.7298594.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[44] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.

[45] EuroStat, "European Environment Agency - Glossary." Accessed: Sep. 16, 2024. [Online]. Available: https://ec.europa.eu/eurostat/