

A comparative analysis of GPUs, TPUs, DPUs, and QPUs for deep learning with python

Ayoub Allali, Zineb El Falah, Ayoub Sghir, Jaafar Abouchabaka, Najat Rafalia

Department of Computer Science, Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco

Article Info

Article history:

Received Jun 19, 2024

Revised Nov 9, 2024

Accepted Nov 24, 2024

Keywords:

Data processing units

Deep learning

Graphics processing units

Python

Quantum processing units

Tensor processing units

ABSTRACT

In the rapidly evolving field of deep learning, the computational demands for training sophisticated models have escalated, prompting a shift towards specialized hardware accelerators such as graphics processing units (GPUs), tensor processing units (TPUs), data processing units (DPUs), and quantum processing units (QPUs). This article provides a comprehensive analysis of these heterogeneous computing architectures, highlighting their unique characteristics, performance metrics, and suitability for various deep learning tasks. By leveraging python, a predominant programming language in the data science domain, the integration and optimization techniques applicable to each hardware platform is explored, offering insights into their practical implications for deep learning research and application. The architectural differences that influence computational efficiency is examined, parallelism, and energy consumption, alongside discussing the evolving ecosystem of software tools and libraries that support deep learning on these platforms. Through a series of benchmarks and case studies, this study aims to equip researchers and practitioners with the knowledge to make informed decisions when selecting hardware for their deep learning projects, ultimately contributing to the acceleration of model development and innovation in the field.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Ayoub Allali

Department of Computer Science, Faculty of Sciences, Ibn Tofail University

Kenitra, Morocco

Email: ayoub.allali@uit.ac.ma

1. INTRODUCTION

The landscape of deep learning has witnessed a remarkable transformation over the last decade, largely fueled by advancements in computational hardware. As models become increasingly complex, the necessity for efficient and powerful computing resources has never been more apparent. The emergence of specialized hardware accelerators, including graphics processing units (GPUs), tensor processing units (TPUs), data processing units (DPUs), and quantum processing units (QPUs), has heralded a new era in the acceleration of deep learning tasks. These technologies offer substantial improvements in processing speed and energy efficiency, enabling researchers and practitioners to tackle more sophisticated problems and achieve breakthroughs at an unprecedented pace [1].

Corral *et al.* [2] evaluate the energy consumption and performance of medical diagnosis aids implemented on embedded processors, specifically Google's Edge TPU and NVIDIA's Maxwell GPU. Using glaucoma detection from color fundus images as a case study, they demonstrate the feasibility of real-time image segmentation and classification on embedded boards. The study focuses on the energy and timing performance of these systems during optic disc (OD) and cup (OC) segmentation and classification tasks,

showing that the Edge TPU is more energy-efficient and faster compared to the Maxwell GPU for both segmentation and classification.

Huang *et al.* [3] focus on improving the efficiency of hierarchical tucker (HT) tensor learning for large-scale, high-dimensional data using GPU tensor cores. They address key challenges in optimizing tensor learning primitives, implementing HT tensor algorithms on multiple GPUs, and handling data exceeding GPU memory. They present optimized GPU-based tensor operations such as contractions, matricizations, and singular value decomposition (SVD) for big data analysis. Additionally, they introduce an HT tensor layer for deep neural networks and develop a quantum machine learning algorithm based on a tensor-tree structure, achieving significant speedups over existing methods on NVIDIA GPUs.

In the reviewed literature, only GPUs and TPUs were focused on, and this may not be enough in the future, because as the data volume increases, more powerful processing units are required to process this data, and the advantages and disadvantages of each unit were not explained, therefore, this article explains the following: i) energy and time consumption of GPUs, TPUs, DPUs, and QPUs running the same deep learning model; ii) how to integrate python, as a major programming language in deep learning, with these hardware accelerators to improve performance and scalability; iii) advantages and disadvantages of GPUs, TPUs, DPUs, and QPUs; and iv) challenges faced by these units.

2. METHOD

The process began with the creation of a deep learning model in python, which served as the foundation for comparison across different processing units. This model was well-defined and adaptable to multiple frameworks to ensure consistency. The choice of the model could have been something computationally demanding, like a convolutional neural network (CNN) or a transformer, depending on the specific tasks under evaluation. The python code was the core logic of the experiment and was the common factor across all hardware platforms, ensuring that the comparisons focused solely on the performance differences of the processing units rather than variations in model implementation [4].

Once the python code was ready as shown in Figure 1, it was deployed on a GPU. This step used either TensorFlow/PyTorch as the deep learning framework, in conjunction with CUDA, the parallel computing platform and application programming interface (API) model that allowed the GPU to handle tasks at a much faster rate. CUDA enabled the python code to run optimized operations, taking full advantage of the GPU's capability to perform massively parallel computations [5]. The GPU was known for accelerating training and inference tasks, making it ideal for many deep learning applications. This stage measured how quickly the model could be processed on the GPU and analyzed the energy consumption involved [6]–[8].

The model was adapted to run on a TPU, a specialized hardware accelerator designed to handle matrix operations commonly used in deep learning. TensorFlow was the preferred framework for TPUs due to its tight integration and optimization for TPU architecture. The TPUs were known to be highly efficient in training large models, especially for operations like convolution and matrix multiplication. By running the model on a TPU, this phase provided insights into how TPUs compared with GPUs in terms of both speed and energy efficiency, especially for large-scale deep learning tasks [9]–[13].

For the DPU, our project involved setting up a robust data processing environment using VMware to create and manage 10 virtual machines (VMs). Each VM was configured to facilitate the testing of a deep learning model, allowing us to evaluate its performance and scalability in a controlled environment. By leveraging VMware's virtualization capabilities, we ensured efficient resource allocation and isolation among the VMs, enabling parallel processing of data and model training. In this step, the model was adapted for the DPU architecture to evaluate how it performed in real-time inference tasks, providing critical data on energy consumption, speed, and processing efficiency in low-power environments [14]–[17].

QPUs represented a cutting-edge approach to computation, leveraging the principles of quantum mechanics to potentially solve problems exponentially faster than classical systems. In this step, the model was reimaged for quantum processing using quantum machine learning frameworks such as Qiskit, Cirq, or PennyLane [18]–[20]. These platforms allowed the model to be executed on quantum hardware, where qubits were used to perform computations. Quantum machine learning was still in its experimental stages, so this step assessed how well quantum systems handled deep learning models and compared their performance to classical systems. Special attention was paid to the energy usage and time taken for tasks that might have taken significantly longer on classical hardware [14], [21]–[23].

After running the deep learning model on all the different processing units, GPU, TPU, DPU, and QPU, the final step was to analyze the results. This involved collecting data on the time taken to execute the model, the energy consumed by each hardware platform, and the overall model performance in terms of accuracy and speed.

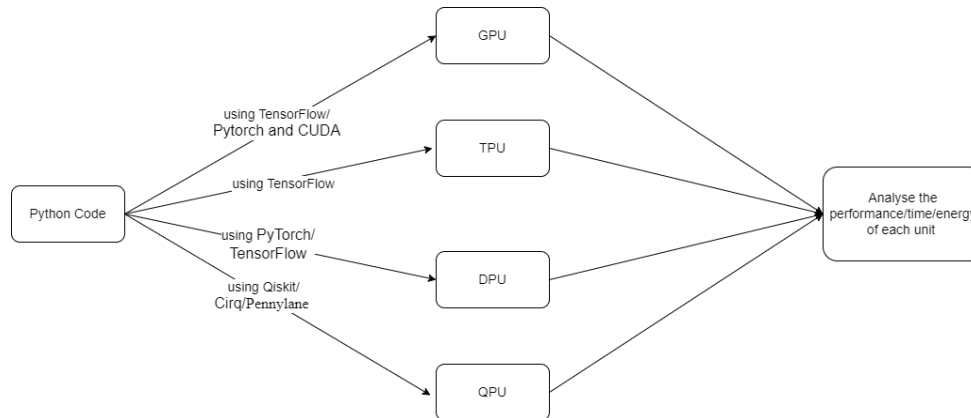


Figure 1. Method

3. RESULTS AND DISCUSSION

The landscape of hardware accelerators for deep learning is rich and varied, with each type of accelerator offering distinct advantages and trade-offs in terms of training time, parallelism, and energy consumption. This comparative analysis delves into GPUs, TPUs, DPUs, and QPUs, highlighting how these technologies stack up against each other in these critical areas.

3.1. Training time

Table 1 shows the timing performance of training the same model using different units. The GPU took about 300 seconds, while the TPU cut this time in half to 150 seconds, reflecting its specialization in deep learning tasks. The DPU was the fastest, completing training in 75 seconds, thanks to its highly optimized architecture for neural network computations. Meanwhile, the Quantum QPU is still in the experimental stage for deep learning, and specific timing data is not yet available. This comparison highlights the superior efficiency of modern specialized DPUs compared to traditional GPUs, while QPU remains at the forefront of cutting-edge innovation [24]–[30].

Table 1. Timing performance of GPU/TPU/DPU/QPU with the same model

	GPU	TPU	DPU	QPU
Time (seconds)	≈ 300s	≈ 150s	≈ 75s	Still in the experimental phase for deep learning

3.2. Energy consumption

The energy consumed by each processing unit (GPU, TPU, DPU, QPU) during deep learning model training depends on several factors, such as the specific hardware, workload, and the duration of training. The total energy consumed E (in Joules or kilowatt-hours) can be estimated using the following formula:

$$E = P * T \quad (1)$$

Where E is the total energy consumed, P is the average power consumption (in watts), T is the time of operation (in seconds or hours) [18], [31]. Table 2 describes the energy consumption in joules per hour for each unit.

Table 2. Comparison table

Unit	Power (W)	Energy for 1 hour (Joules/MJ)
GPU	400	1,440,000 J (1.44)
TPU	300	1,080,000 J (1.08)
DPU	75	270,000 J (0.27)
QPU	50	180,000 J (0.18)

High-performance GPUs can have a power consumption of 250 W to 400 W, and training time varies depending on model complexity, data, and hardware but $T=1$ hour (3,600 seconds) for simplicity is assumed [32].

$$E = P * T = 400 * 3600 = 1,440,000 \text{ Joules (1.44 MJ)} \tag{2}$$

TPUs are more power-efficient for large-scale computations. A TPUv3 pod can have an average power consumption of around 200W to 300W per TPU core [33].

$$E = P * T = 300 * 3600 = 1,080,000 \text{ Joules (1.08 MJ)} \tag{3}$$

DPUs like the NVIDIA BlueField-2 generally consume around 50W to 75W depending on the workload. However, since DPUs are typically not used for model training, their energy consumption in this context would relate to data handling tasks [34].

$$E = P * T = 75 * 3600 = 270,000 \text{ Joules (0.27 MJ)} \tag{4}$$

QPUs are highly experimental, and power usage varies significantly, QPUs in research environments can consume anywhere from 10W to 50W or more, but they are generally not used for prolonged deep learning training [35]–[37].

$$E = P * T = 50 * 3600 = 180,000 \text{ Joules (0.18 MJ)} \tag{5}$$

3.3. Parallelism

Parallelism is a measure of how effectively a computing system can perform multiple operations simultaneously. GPUs excel in this domain, with architectures designed to handle thousands of threads in parallel. This capability makes them exceptionally suited for the parallel nature of deep learning computations. TPUs also offer significant parallel processing capabilities, especially optimized for high-volume, low-precision arithmetic operations common in deep learning algorithms [38]–[43].

DPUs contribute to parallelism indirectly by offloading and accelerating networking, security, and I/O operations, which in turn enables more efficient use of computing resources for parallel computations in deep learning workloads. QPUs, on the other hand, offer a form of parallelism that is qualitatively different from classical systems, using quantum entanglement and superposition to perform a vast number of calculations simultaneously, although their application in deep learning is still emerging [44].

3.4. Integration and optimization

The integration of these accelerators into deep learning workflows varies widely. GPUs are supported by a mature ecosystem of tools and libraries, such as CUDA for NVIDIA GPUs, making them relatively straightforward to integrate into existing deep learning frameworks. TPUs are also well-supported, particularly within Google’s ecosystem, with TensorFlow offering seamless integration. DPUs require more specialized integration efforts, focusing on the optimization of data center operations rather than direct acceleration of deep learning models. QPUs, being at the forefront of computational research, currently require highly specialized knowledge to use effectively in deep learning applications [45], [46].

3.5. Challenges

In deep learning, each specialized processing unit GPU, TPU, DPU, and QPU faces distinct challenge describes in Table 3 GPUs are powerful for parallel computations but struggle with high energy consumption, memory bandwidth limitations, and inefficiency in small models. TPUs, designed for tensor operations, offer strong performance but are limited by their lack of flexibility, steep learning curve, and dependence on large batch sizes. DPUs, while useful for distributed systems by handling data and network tasks, have niche use cases, complex software integration, and limited maturity. QPUs offer theoretical speed-ups for specific tasks but face major hurdles, such as hardware immaturity, integration with classical systems, and algorithmic development, making them less practical for current deep learning needs [47], [48].

Table 3. Challenges of each unit

Unit	GPU	TPU	DPU	QPU
Major challenges	– High energy consumption	– Limited flexibility	– Niche use cases	– Hardware immaturity
	– Memory bandwidth bottlenecks	– Programming complexity	– Software ecosystem limitations	– Algorithmic limitations
	– Scalability issues	– Memory bottlenecks	– Integration overhead	– Classical quantum integration issues
	– Inefficiency with small models	– Cloud-only availability	– Cost	– Limited use cases
	– High cost	– Batch size sensitivity	– Maturity of technology	– High cost

Lastly, selecting the right hardware accelerator for a deep learning project involves a careful consideration of the specific requirements and constraints of the project, including computational efficiency, parallelism, energy consumption, and the ease of integration. As this field continues to evolve rapidly, staying informed about the latest developments in hardware accelerators will be crucial for maximizing the performance and efficiency of deep learning applications.

4. CONCLUSION

While GPUs, TPUs, DPUs, and QPUs each bring significant potential to accelerate deep learning tasks, they also come with unique challenges that must be considered based on the specific use case. GPUs remain the most versatile and widely used, but their energy demands and scalability issues can be limiting. TPUs offer specialized performance advantages, particularly for large-scale models, but are less flexible and more difficult to program. DPUs serve niche roles in distributed systems but require complex integration, and QPUs, though promising, are still in the experimental stage and not yet practical for general-purpose deep learning. The choice of hardware ultimately depends on balancing these trade-offs with the specific needs of the deep learning application.




REFERENCES

- [1] N. Shah, L. I. G. Olascoaga, S. Zhao, W. Meert, and M. Verhelst, "DPU: DAG processing unit for irregular graphs with precision-scalable posit arithmetic in 28 nm," *IEEE Journal of Solid-State Circuits*, vol. 57, no. 8, pp. 2586–2596, 2022, doi: 10.1109/JSSC.2021.3134897.
- [2] J. M. Rodríguez Corral, J. Civit-Masot, F. Luna-Perejón, I. Díaz-Cano, A. Morgado-Estévez, and M. Domínguez-Morales, "Energy efficiency in edge TPU vs. embedded GPU for computer-aided medical imaging segmentation and classification," *Engineering Applications of Artificial Intelligence*, vol. 127, Jan. 2024, doi: 10.1016/j.engappai.2023.107298.
- [3] H. Huang, X. Y. Liu, W. Tong, T. Zhang, A. Walid, and X. Wang, "High performance performance hierarchical tucker tensor learning using GPU tensor cores," *IEEE Transactions on Computers*, vol. 72, no. 2, pp. 452–465, 2023, doi: 10.1109/TC.2022.3172895.
- [4] S. C. Magalhães, F. N. dos Santos, P. Machado, A. P. Moreira, and J. Dias, "Benchmarking edge computing devices for grape bunches and trunks detection using accelerated object detection single shot multibox deep learning models," *Engineering Applications of Artificial Intelligence*, vol. 117, 2023, doi: 10.1016/j.engappai.2022.105604.
- [5] S. Laue, M. Blacher, and J. Giesen, "Optimization for classical machine learning problems on the GPU," in *Proceedings of the 36th AAAI Conference on Artificial Intelligence, AAAI 2022*, 2022, vol. 36, pp. 7300–7308, doi: 10.1609/aaai.v36i7.20692.
- [6] J. S. Lerat, S. A. Mahmoudi, and S. Mahmoudi, "Single node deep learning frameworks: Comparative study and CPU/GPU performance analysis," *Concurrency and Computation: Practice and Experience*, vol. 35, no. 14, 2023, doi: 10.1002/cpe.6730.
- [7] S. Thomas, M. L. Bowers, and W. Liao, "GPU computing and its applications in AI and deep learning," *Proceedings of the West Virginia Academy of Science*, vol. 90, no. 1, Apr. 2018, doi: 10.55632/pwvas.v90i1.408.
- [8] D. Zhang, Y. Luo, Y. Wang, X. Kui, and J. Ren, "BatOpt: Optimizing GPU-based deep learning inference using dynamic batch processing," *IEEE Transactions on Cloud Computing*, vol. 12, no. 1, pp. 174–185, 2024, doi: 10.1109/TCC.2024.3350561.
- [9] X. Shi, X. Peng, L. He, Y. Zhao, and H. Jin, "Waterwave: A GPU memory flow engine for concurrent DNN training," *IEEE Transactions on Computers*, vol. 72, no. 10, pp. 2938–2950, 2023, doi: 10.1109/TC.2023.3278530.
- [10] Y. Wang *et al.*, "DRLCAP: Runtime GPU frequency capping with deep reinforcement learning," *IEEE Transactions on Sustainable Computing*, vol. 9, no. 5, pp. 712–726, Sep. 2024, doi: 10.1109/TSUSC.2024.3362697.
- [11] X. Zhang, "Mixtran: An efficient and fair scheduler for mixed deep learning workloads in heterogeneous GPU environments," *Cluster Computing*, vol. 27, no. 3, pp. 2775–2784, 2024, doi: 10.1007/s10586-023-04104-9.
- [12] L. Denis, R. Royen, Q. Bolsée, N. Vercheval, A. Pižurica, and A. Munteanu, "GPU rasterization-based 3D LiDAR simulation for deep learning," *Sensors*, vol. 23, no. 19, 2023, doi: 10.3390/s23198130.
- [13] R. L. Castro, D. Andrade, and B. B. Fraguera, "STuning-DL: Model-driven autotuning of sparse GPU kernels for deep learning," *IEEE Access*, vol. 12, 2024, doi: 10.1109/ACCESS.2024.3402326.
- [14] A. Ravikumar, H. Sriraman, P. M. Sai Saketh, S. Lokesh, and A. Karanam, "Effect of neural network structure in accelerating performance and accuracy of a convolutional neural network with GPU/TPU for image analytics," *PeerJ Computer Science*, vol. 8, Mar. 2022, doi: 10.7717/peerj-cs.909.
- [15] Z. Ye *et al.*, "Deep learning workload scheduling in GPU datacenters: A survey," *ACM Computing Surveys*, vol. 56, no. 6, pp. 1–38, Jun. 2024, doi: 10.1145/3638757.
- [16] Z. Peng, J. Du, and Y. Qiao, "Design of GPU network-on-chip for real-time video super-resolution reconstruction," *Micromachines*, vol. 14, no. 5, May 2023, doi: 10.3390/mi14051055.
- [17] J. Raymond *et al.*, "Hybrid quantum annealing for larger-than-QPU quantum annealing for larger-than," *ACM Transactions on Quantum Computing*, vol. 4, no. 3, pp. 1–30, Sep. 2023, doi: 10.1145/3579368.
- [18] J. Vázquez-Pérez, C. Piñeiro, J. C. Pichel, T. F. Pena, and A. Gómez, "QPU integration in OpenCL for heterogeneous programming," *The Journal of Supercomputing*, vol. 80, no. 8, pp. 11682–11703, May 2024, doi: 10.1007/s11227-023-05879-9.
- [19] T. Tan and G. Cao, "Thermal-aware scheduling for deep learning on mobile devices with NPU," *IEEE Transactions on Mobile Computing*, vol. 23, no. 12, pp. 10706–10719, Dec. 2024, doi: 10.1109/TMC.2024.3379501.
- [20] A. Martín-Martín *et al.*, "Hardware implementations of a deep learning approach to optimal configuration of reconfigurable intelligence surfaces," *Sensors*, vol. 24, no. 3, Jan. 2024, doi: 10.3390/s24030899.
- [21] A. Morningstar *et al.*, "Simulation of quantum many-body dynamics with tensor processing units: Floquet prethermalization," *PRX Quantum*, vol. 3, no. 2, May 2022, doi: 10.1103/PRXQuantum.3.020331.
- [22] O. Huang and M. L. Palmeri, "TPU based deep learning image enhancement for real-time point-of-care ultrasound," *IEEE Transactions on Computational Imaging*, vol. 10, pp. 461–468, 2024, doi: 10.1109/TCI.2024.3372445.




- [23] H. Choi, B. H. Lee, S. Y. Chun, and J. Lee, "Towards accelerating model parallelism in distributed deep learning systems," *PLOS ONE*, vol. 18, no. 11, Nov. 2023, doi: 10.1371/journal.pone.0293338.
- [24] A. Erciyas and N. Barişçi, "A meta-analysis on diabetic retinopathy and deep learning applications," *Multimedia Tools and Applications*, vol. 83, no. 19, pp. 57429–57448, Dec. 2023, doi: 10.1007/s11042-023-17784-7.
- [25] F. G. Tan, "The impact of deep learning and transfer learning algorithms on drone detection performance, (In Turkish: Derin öğrenme ve öğrenme aktarımı algoritmalarının drone algılama performansı üzerine etkisi)," *Gazi Journal of Engineering Sciences*, vol. 9, no. 4, pp. 1–13, Dec. 2023, doi: 10.30855/gmbd.0705S01.
- [26] G. Akkad, A. Mansour, and E. Inaty, "Embedded deep learning accelerators: A survey on recent advances," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 5, pp. 1954–1972, May 2024, doi: 10.1109/TAI.2023.3311776.
- [27] Y. Wu *et al.*, "Elastic deep learning in multi-tenant GPU clusters," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 1, pp. 144–158, Jan. 2022, doi: 10.1109/TPDS.2021.3064966.
- [28] H. Choi and J. Lee, "Efficient use of GPU memory for large-scale deep learning model training," *Applied Sciences*, vol. 11, no. 21, Nov. 2021, doi: 10.3390/app112110377.
- [29] M. Pandey *et al.*, "The transformational role of GPU computing and deep learning in drug discovery," *Nature Machine Intelligence*, vol. 4, no. 3, pp. 211–221, Mar. 2022, doi: 10.1038/s42256-022-00463-x.
- [30] Z. Chen, X. Zhao, C. Zhi, and J. Yin, "DeepBoot: Dynamic scheduling system for training and inference deep learning tasks in GPU cluster," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 9, pp. 2553–2567, Sep. 2023, doi: 10.1109/TPDS.2023.3293835.
- [31] Y. Liao, J. Wu, W. Lu, X. Li, and G. Yan, "DPU-direct: Unleashing remote accelerators via enhanced RDMA for disaggregated datacenters," *IEEE Transactions on Computers*, vol. 73, no. 8, pp. 2081–2095, Aug. 2024, doi: 10.1109/TC.2024.3404089.
- [32] B. Chopra, "Enhancing machine learning performance: The role of GPU-based AI compute architectures," *Journal of Knowledge Learning and Science Technology*, vol. 3, no. 3, pp. 29–42, Mar. 2024, doi: 10.60087/jklst.vol3.n3.p40.
- [33] Z. Pan, F. Zhang, H. Li, C. Zhang, X. Du, and D. Deng, "G-SLIDE: A GPU-based sub-linear deep learning engine via LSH sparsification," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 11, pp. 3015–3027, 2021, doi: 10.1109/TPDS.2021.3132493.
- [34] P. Maillard *et al.*, "Radiation tolerant deep learning processor unit (DPU) based platform using Xilinx 20nm Kintex UltraScale™ FPGA," *IEEE Transactions on Nuclear Science*, vol. 70, no. 4, pp. 714–721, Apr. 2023, doi: 10.1109/TNS.2022.3216360.
- [35] C. Li, R. Xu, Y. Lv, Y. Zhao, and W. Jing, "Edge real-time object detection and dpu-based hardware implementation for optical remote sensing images," *Remote Sensing*, vol. 15, no. 16, Aug. 2023, doi: 10.3390/rs15163975.
- [36] Z. Wang, C. Wang, and L. Wang, "DPUBench: An application-driven scalable benchmark suite for comprehensive DPU evaluation," *BenchCouncil Transactions on Benchmarks, Standards and Evaluations*, vol. 3, no. 2, Jun. 2023, doi: 10.1016/j.tbench.2023.100120.
- [37] R. Wang, X. Yu, Q. Wu, C. Yi, P. Wang, and D. Niyato, "Efficient deployment of partial parallelized service function chains in CPU+DPU-based heterogeneous NFV platforms," *IEEE Transactions on Mobile Computing*, vol. 23, no. 10, pp. 9090–9107, Oct. 2024, doi: 10.1109/TMC.2024.3357796.
- [38] X. Geng, H. Zhang, Z. Zhao, and H. Ma, "Interference-aware parallelization for deep learning workload in GPU cluster," *Cluster Computing*, vol. 23, no. 4, pp. 2689–2702, Dec. 2020, doi: 10.1007/s10586-019-03037-6.
- [39] A. Kalantar, Z. Zimmerman, and P. Brisk, "FPGA-based acceleration of time series similarity prediction: From cloud to edge," *ACM Transactions on Reconfigurable Technology and Systems*, vol. 16, no. 1, pp. 1–27, Mar. 2023, doi: 10.1145/3555810.
- [40] S.-O. Park, O. Kwon, Y. Kim, S. K. Cha, and H. Yoon, "Mind control attack: Undermining deep learning with GPU memory exploitation," *Computers & Security*, vol. 102, Mar. 2021, doi: 10.1016/j.cose.2020.102115.
- [41] Z. Chen, "RIFLING: A reinforcement learning-based GPU scheduler for deep learning research and development platforms," *Software: Practice and Experience*, vol. 52, no. 6, pp. 1319–1336, Jun. 2022, doi: 10.1002/spe.3066.
- [42] T. T. J. Kiran, "Deep transform learning vision accuracy analysis on GPU using tensor flow," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 9, no. 3, pp. 224–227, Sep. 2020, doi: 10.35940/ijrte.C4402.099320.
- [43] H. Liu, S. Liu, C. Wen, and W. E. Wong, "TBEM: Testing-based GPU-memory consumption estimation for deep learning," *IEEE Access*, vol. 10, pp. 39674–39680, 2022, doi: 10.1109/ACCESS.2022.3164510.
- [44] J. Wei, X. Zhang, L. Wang, and Z. Wei, "Fastensor: Optimise the tensor I/O path from SSD to GPU for deep learning training," *ACM Transactions on Architecture and Code Optimization*, vol. 20, no. 4, pp. 1–25, Dec. 2023, doi: 10.1145/3630108.
- [45] D.-K. Kang, K.-B. Lee, and Y.-C. Kim, "Cost efficient GPU cluster management for training and inference of deep learning," *Energies*, vol. 15, no. 2, Jan. 2022, doi: 10.3390/en15020474.
- [46] D. Vasan, M. Hammoudeh, and M. Alazab, "Broad learning: A GPU-free image-based malware classification," *Applied Soft Computing*, vol. 154, Mar. 2024, doi: 10.1016/j.asoc.2024.111401.
- [47] W. Gao, Z. Ye, P. Sun, T. Zhang, and Y. Wen, "UniSched: A unified scheduler for deep learning training jobs with different user demands," *IEEE Transactions on Computers*, vol. 73, no. 6, pp. 1500–1515, Jun. 2024, doi: 10.1109/TC.2024.3371794.
- [48] P. V. Lakshmi, T. A. S. Srinivas, A. D. Donald, S. Abida, I. Dwaraka, and Srihith, "GPU magic: Turbocharging your machine learning models," *Computer Graphics and Multimedia Technology, SAD*.

BIOGRAPHIES OF AUTHORS






Ayoub Allali    born on August 1, 1997, in Sidi Kacem, Morocco, has pursued an academic and research career in the fields of computer science and big data. He completed his Bachelor's degree in Mathematical and Computer Sciences from the Department of Computer Science at the Faculty of Sciences, Ibn Tofail University in Kenitra, Morocco, in 2019. Continuing his education at the same institution, Ayoub earned a Master's degree in Big Data and Cloud Computing in 2021. Since 2022, he has been a doctoral student at the Computer Research Laboratory (LaRI) within the Department of Computer Science at the Faculty of Sciences, Ibn Tofail University. His research interests are focused on big data and deep learning, indicating a strong commitment to advancing knowledge in these critical areas of data science. He can be contacted at email: ayoub.allali@uit.ac.ma.






Zineb El Falah    born on February 24, 1997, in Meknes, Morocco, has dedicated her academic and research career to the fields of computer science and big data. She obtained her Master's degree in Big Data and Cloud Computing in 2020. Since 2021, she has been a Ph.D. student at the Computer Science Research Laboratory (LaRi) in the Faculty of Science at Ibn Tofail University. Her research focuses on data analysis and decision-making in big data, utilizing artificial intelligence, machine learning, and deep learning. This demonstrates her strong commitment to advancing knowledge in these critical areas of data science. She can be contacted at email: zineb.elfalah@uit.ac.ma.






Ayoub Sghir    was born in 1996 in Sale. He received his Master's degree in computer science, big data cloud computing from Ibn Tofail University, Kenitra, Morocco. He is a Ph.D. student in Computer Research Laboratory (LaRI) at Ibn Tofail. His research interests include big data, data storage, cloud computing, and distributed computing. He can be contacted at email: ayoub.sghir@uit.ac.ma.



Jaafar Abouchabaka    born on February 27, 1968, in Guersif, Morocco, is a distinguished academic with a prolific career in mathematics and computer science. He embarked on his educational journey at Kadi Ayad University in Marrakech, Morocco, where he obtained his Bachelor's degree in Fundamental Mathematics from the Department of Mathematics, Faculty of Sciences, in 1992. He furthered his studies at Mohammed V University in Rabat, Morocco, earning a Postgraduate Diploma (DEA) in Mathematics in 1994, and later a Doctorate in Computer Sciences Applied to Mathematics from the Department of Computer Science in 2001. Since 2005, he has held the position of Professor in the Department of Computer Sciences at Ibn Tofail University in Kenitra, Morocco. His research interests are broad and impactful, focusing on concurrent and parallel programming, distributed systems, artificial intelligence, and big data. Over the course of his career, he has contributed significantly to the academic community, authoring more than 70 papers and completing 2 theses. He can be contacted at email: jaafar.abouchabaka@uit.ac.ma.



Najat Rafalia    born on November 4, 1968, in Kenitra, Morocco, has established herself as a prominent figure in the fields of computer science and applied mathematics. Her academic journey began at Mohammed V University in Rabat, Morocco, where she received her Bachelor's degree in Applied Mathematics from the Department of Mathematics, Faculty of Sciences, in 1992. She continued at the same institution to earn a Postgraduate Diploma (DEA) in Computer Science in 1994 and later a Doctorate in Computer Sciences from the Department of Computer Science in 1997. Since 1997, she has served as a Professor in the Department of Computer Sciences at Ibn Tofail University. Further advancing her expertise, she had her postdoctoral thesis in 2013 and completed her third Doctorate in Computer Sciences at Ibn Tofail University in Kenitra, Morocco, in 2017. Her research interests are deeply rooted in big data analytic, artificial intelligence and its applications, internet of things, distributed systems, multi-agent systems, concurrent and parallel programming. Throughout her career, she has made significant contributions to her field, authoring more than 70 papers and completing 3 theses. She can be contacted at email: najat.rafallia@uit.ac.ma.