Benchmarking Machine Learning Inference in FPGA-based Accelerated Space Applications

Amir Raoofy*, Gabriel Dax*,
Max Ghiglione§, Martin Langer†, Carsten Trinitis*, Martin Werner*, Martin Schulz*
TU Munich* Airbus Defence and Space GmbH, Munich§ Orbital Oracle Technologies GmbH, Munich†
{amir.raoofy, gabriel.dax,
carsten.trinitis, martin.werner, martin.w.j.schulz}@tum.de max.ghiglione@airbus.com martin.langer@ororatech.com

Abstract

Special requirements of space missions, including limited energy budgets and radiation tolerance, impose strict operational conditions on on-board data processing system. Consequently, deploying *Machine Learning (ML) inference* to data processing systems in satellites introduces architectural and practical challenges. In this position paper, we discuss these challenges of using *FPGAs* for the acceleration of ML inference as a main trend in the evolution of on-board data processing. We envision the rising need for the development of benchmarks and key performance indicators to characterize space-enabled FPGA-based solutions for accelerating ML inference for satellite-based platforms.

Keywords: Machine Learning Inference, FPGA-based Acceleration, Inference Benchmarking, In-Orbit Data Processing.

1 Introduction

With continuing advances of modern ML technologies such as *Deep Neural Networks (DNNs)* and especially *Convolutional Neural Networks (CNNs)*, space applications can adopt and take advantage of the capabilities of models and algorithms. The European Space Agency (ESA) is interested in deploying *ML inference workloads* to on-board data processing systems for various applications such as, e.g., Earth observation [8] or optical guidance, navigation and control (GN&C) [1]. In these scenarios, ML models are trained on the ground and then deployed to satellite-based platforms for *on-board* inference.

One of the main reasons for deploying such on-board ML inference is the limited bandwidth between satellites and ground stations. As a consequence, it is typically necessary to limit satellites to take and transmit detailed pictures of selected regions only with a subset of sensor modes, as it is impossible to cache and downstream all available observation data. With on-board inference capabilities enables a *real-time* decision on what to focus on and what to stream down to Earth allowing innovative data-driven observation scenarios. This enables us to increase the "scientific content of the downloaded data" [8], while fitting within the practical transition limits.

However, the power efficiency demands of satellite missions are extreme not only due to the fact that the energy needs to be collected from solar panels, but also that the heat generated needs to be radiated away from the satellite, which is hard to accomplish due to the vacuum in space despite low temperatures.

In this context, there is a clear need for the development of ML inference benchmarks for various space applications and models, that focus on characterizing real-time performance, power efficiency.

Despite little attention in the past [10], recent years have seen a rising attention on Field-Programmable Gate Arrays (FPGAs) for the acceleration of various ML workloads, in particular for space applications, due to the increasing availability of both radiation-tolerant devices and Commercial-Off-The-Shelf (COTS) solutions. Specifically, research shows promising results for the deployment of ML inference on FPGAs [12]. Moreover, the unique features of these devices, e.g. low power consumption and HW (re)programmability, have drawn the attention of space industry practitioners to investigate these accelerators further. However, despite the rising attention and the increasing use of FPGA-based inference accelerators, there is little work on benchmarking ML inference on such radiation-tolerant, space-capable HW with their specific resource limitations as well as specific model properties.

In summary, there is a need to define the key performance indicators for space ML applications, together with the design and implementation of a benchmark system covering these indicators. This will enable the community to systematically evaluate the trade-offs of various approaches. This position paper focuses on such techniques and benchmarking systems for FPGA-based inference in space applications, targeting space-enabled HW and SW, filling a gap for both for a particular program funded by ESA, but also providing a long-term tool for similar deployment scenarios in the wider community.

2 Emerging Inference Accelerators and ML Inference Development Workflow

To cope with the requirements of future space missions, the space industry tends to design and exploit reconfigurable and hybrid architectures for data processing, exploiting multiple embedded compute nodes. This reconfigurable and hybrid data processing promises in-orbit deployment of a wider range of science and intelligence through the use of ML inference. CPUs, GPUs, DSPs, and reconfigurable FPGAs are

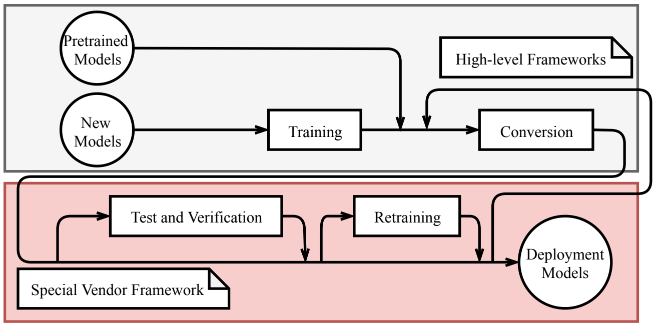

Figure 1. Illustration of the workflow for the development of FPGA-based inference accelerator using high-level and vendor tools for automatic HW design and model quantization.

the drivers for the next generations of on-board data processing [1, 4, 6]. However, in the context of space missions FPGAs are preferable mainly due to low power consumption, and availability of radiation-tolerant solutions. For example, NASA's SpaceCube v2.0 uses radiation-hardened reconfigurable FPGA-based acceleration in data processing systems [9]. Another example is the appearance of systems based on radiation-mitigated COTS solutions, such as MPSoCs and ACAPs that provide hybrid and flexible compute paradigms, and are getting more attention by the space industry [6]. Thus, it is important to design inference benchmarks on top of these solutions to characterize their performance.

Recent developments have significantly reduced the effort for the deployment of ML inference, through the automation of the HW design process using supplementary development kits provided by the vendors (see Section 1), specifically for commercial COTS solutions. They provide versatile platforms that satisfy most of the data processing requirements in future space missions and hence offer suitable target platforms for the realization of the ML inference benchmarks for space applications. Figure 1 illustrates a workflow for the development of ML inference solutions. This workflow includes the training phase to cope with the in-progress and continuous development of new modelsand fine-tuning of existing models for space applications. This phase is performed using high-level tools, e.g., Tensorflow, Caffe, and Pytorch on HPC systems, and exploits parallel training on multiple CPU- or GPU-based nodes to cope with the high computational needs as well as with the large amount of training data. After training, the models are pruned and converted to exploit simpler data formats, e.g., *float16*, and after verification (e.g., offline testing) of the converted models and calibration in a so-called retraining phase, models are linked to pre-synthesized soft IPs designed by the vendors or customized IPs to be deployed to the board for acceleration.

Training and conversion can be executed on any platform that provides a backend for high-level tools, like e.g. Tensorflow. However, the rest of the workflow requires vendor support. Nowadays, vendors implement this workflow in proprietary toolsets for their specific HW platforms. For instance, Xilinx implements this workflow in Vitis AI[1], and Intel implements it in OpenVINO[2]. On the other hand, there are also open-source solutions, e.g., ChaiDNN for specific targets. However, most of the solutions typically include the same steps. Aside from the above workflow, experimental tools, such as Brevitas [7] and FINN [12] offer a different workflow (Brevitas uses quantized models at the point of training). However, a benchmarking system based on the workflow in Figure 1 covers that as well.

Although using the high-level tools for HW design might have downsides, it significantly boosts the prototyping speed for benchmarks. Moreover, it brings in benchmarks with a broader range of deployable ML inference models for space applications on the FPGA-based accelerators.

3 Challenges of In-Orbit ML Inference

Space missions can benefit from ML inference for various use cases, including optical Guidance, Navigation and Control (GN&C) [1] or satellite imaging [4]. However, deploying ML inference to on-board systems requires the consideration of various architectural and system design challenges, including: (1) limited in-orbit compute and memory resources, (2) fault mitigation, (3) HW reconfigurability of the inference engine (4) strict energy and heat budget.

Limited on-board resources. The main driver for on-board processing in Earth observation satellites is the limited downstream capacity. Currently, data is streamed mainly over the poles and satellites trade-off a low-resolution continuous capture of a pass with a selective acquisition in very high resolution, but with limited spatial extent [4].

Being able to deploy a data-driven and machine-learning enabled subsystem in space in order to decide what to capture and what to transmit down to Earth would be a great advantage and lead to more valuable information being transmitted to Earth.

However, especially high-resolution sensors create high data volumes and, thus, need very specialized computational systems given the limited energy budget and real-time constraints. Furthermore, the data acquired on the satellite is comparably far away from analysis-ready data limiting the applicability of ML models to very complex raw data.

This suggests the benefits in the more extensive use of the on-board parallel HW and accelerators for inference processing both for preprocessing as well as for ML inference. Consequently, we further observe the deployment of accelerators with low-power consumption, such as Intel Movidius NCS, and Xilinx Virtex-5QV FPGAs, in conjunction with multiprocessors and DSPs [1]. This yields a hybrid parallel computing paradigm suitable for the deployment of ML and non-trivial

[1] Xilinx's development platform for AI inference, link.

[2] Intl's model optimization and deployment toolkit, link.

preprocessing and signal processing steps for the on-board data processing.

Fault-mitigation. For in-orbit ML, both the input data and the processing platform are exposed to faults. *radiation-hardened* HW can be used at the cost of reduced performance, and researchers are now evaluating radiation-hardened FPGAs, as well as *non-radiation-hardened* accelerators, e.g., GPUs, and COTS solutions, and are considering various SW and HW-based mitigation approaches [13], where fault mitigation (e.g., scrubbing and modular redundancy) is embedded into the AI inference engine. Also, recently more comprehensive benchmarks are developed for the comparison of various parallel workloads on radiation-hardened multi-core space processors [3]. However, the benchmarks in this work do not cover ML inference and HW accelerators. On the other hand, the existing benchmark coverage for ML inference for FPGA-accelerated solutions is limited, and within this context, it is essential to consider fault tolerant FPGA-accelerated solutions and various approaches of fault mitigation.

Reconfigurability. Another architectural feature for future on-board data processing systems is HW reconfigurability [6], where inference HW can be adapted according to the mission requirements. Moreover, reconfiguration has the additional advantage in space applications enabling the model fine-tuning using the data even within missions. In the specific case of Earth observation or anomaly detection, by having fine-tuned models, e.g., seasonal effects can be taken into account in a better way. In this context, FPGAs promise efficient and reconfigurable systems to adapt functionality in various missions [2].

Power/energy/heat efficiency. ML inference, especially for deep convolutional models, is typically compute- and therefore power/energy-hungry. On the other side, spacecrafts are extremely limited in terms of energy due to the limited size of solar panels. Additionally, the on-board data processing systems have to be cooled without fans, which limits the acceptable dissipated heat. Especially on small satellites for Earth observation, overall payload power is operated under 100 Watts. As a consequence the use of low-power accelerators in space missions becomes essential. Benchmarks must therefore be able to evaluate the overall power consumption trade-offs among different solutions. For instance, running the additional inference module consumes more power, while the communication module requires less power to transmit only intelligently-reduced or a *selected subset* of representations instead of raw data. Consequently, benchmarks should be able to evaluate the *overall* power consumption footprints to characterize on-board ML inference solutions properly.

Integration into industry workflow. As FPGAs are conventionally programmed using manual low-level RTL or HLS designs, which require expert knowledge in HW design. In the case of ML, the target tasks are very specific, but they require an additional level of expert knowledge in ML. In addition, due to the continuous and rapid evolution of modern ML models, specifically CNNs, low-level manual design is very cumbersome and inefficient as it requires manual adaptation of the designs for each model. As a result, manual design is very expensive and only performed for special tasks. This has motivated researchers and vendors to investigate the automation of the HW design process for ML workloads on FPGAs, which significantly reduces the cost of the development while allowing for flexible HW designs. This automation is also helpful for designing optimal inference accelerators, and enabling evaluation of trade-offs for different design decisions, and realizing flexible and adaptable designs, e.g., according to the requirements of different missions.

4 Workloads for Space Applications

Inference benchmarks should cover a *diverse set of algorithms* including feature extraction, object detection, classification, tracking, and change detection. This ensures that various space use-cases (see Section 1) are represented in benchmarks. Another aspect corresponding to the representative workloads is the diversity of ML models. This diversity ensures the coverage of various models with different computational complexity and memory requirements. As many space applications [5] often adopt and use well-stablished models, deployment of various classical convolutional models with different network architectures, from light-weight models such as MobileNetV2 (14MB) to more complex ones such as VGG19 (549MB), need to be evaluated. Moreover, space applications rely on different datasets than classical ML applications. So benchmarks need to rely on publicly available and standardized space datasets [5, 11, 14] and splits [5] as well as models to ensure wide reproducibility of any achieved results.

Further, the benchmarks should cover various types of inference scenarios [10]. These scenarios represent various schemes in arrival and dispatching of queries from the applications to the inference module, e.g., streams of *single queries*, *batch* queries, latency-insensitive streams of *batch queries*, latency-sensitive *random single queries*. These various scenarios are diverse and cover the various needs of different space applications, and can be implemented in the benchmark applications as an inference load generator component and, e.g., be executed on the CPUs of MPSoCs.

Due to practical in-orbit limitations and the possible need for redundancy for fault mitigation requirements, reduced-precision computation is seen as one valuable approach and needs to be covered in benchmarks. Here, FPGAs offer flexible reduced-precision models. However, practical domain-specific accuracy measures and performance metrics for various scenarios, e.g., Earth observation, need to be established to characterize the tradeoffs. In this context various approaches in reduced-precision models, i.e., with reduced-precision weights and activations [12], and lossy data representations, covering

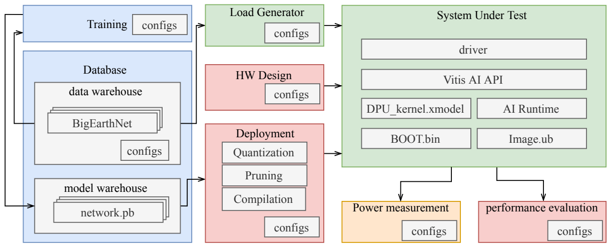

Figure 2. Illustration of inference benchmarking system for FPGA-based systems. We color-coded various subsystems of the benchmarking system.

a wide range of accuracy and performance trade-offs, in combination with fault mitigation approaches need to be evaluated.

5 Benchmarking Systems and Practical Challenges

Figure 2 illustrates a possible realization of the on-board inference benchmarking system. This system is designed based on MLPerf inference benchmarking [10] method and is augmented with the Xilinx Vitis AI system[3], and a power measurement component. It uses an HPC Backend for data preparation and model training, which can benefit from distributed and accelerated computing. The benchmarking system further consists of a Design and Deployment subsystem for HW design and preparation of ML inference models. This subsystem can be realized in a powerful workstation or in an HPC backend exploiting data parallelism for the execution of the benchmark. Performance evaluation component helps to implement performance metrics, e.g., domain-specific accuracy, FPGA resource utilization and efficiency. The Test Unit includes the target board (here Xilinx MPSoC) and executes the *Load Generator* and benchmark *driver* while offloading parts of the computation to the on-board FPGA. The load generator is used to simulate the loads for radiation testing as well as various query scenarios. Power and energy measurements are performed in an external subsystem. This system enables exploration of the design space of inference deployment to FPGAs (see Section 5).

The development of reliable and reproducible FPGA-based acceleration using this benchmakring system requires dealing with a number of practical challenges:

Performance bottleneck in feeding the FPGA chip. The benchmarking system in Figure 2 relies on a load (query) generator and a database to simulate the diverse query scenarios discussed in Section 4. However, the deployment of a load generator on the low-end on-board CPU (e.g., in MPSoC COTS) might not saturate the FPGA chip, resulting in suboptimal benchmarking for batch scenarios. Further, on-board load generation might affect the board's power footprint, which is not desired for benchmarking. As a result, the benchmarking system might need to deploy an external load generator that interacts with the COTS board. In this way, both passive payloads receiving data from sensors and active processing units fetching data from the mass memory of the spacecraft can be evaluated.

Consistent power measurement. As discussed in Section 5, the inference benchmarks should be able to evaluate the energy footprint for various inference tasks. However, the evaluation of static and dynamic on-chip and off-chip power measurements can be a difficult task, and vendors rely on tools that exploit power consumption estimators and analyzers (such as Xilinx XPE). For this reason, deriving energy footprints for the boards using vendor-agnostic external power measurement units to evaluate the power swings during benchmarking is a more straightforward and consistent approach. This requires the deployment of additional external infrastructure for power measurement during the installation of the HW boards.

Hardware acceleration approaches. Acceleration of ML inference using FPGAs is an active topic of research and development. Various approaches, e.g., Vitis AI using DPUs, DNNDK, and FINN need to be evaluated, which requires a detailed study of vendor IP designs. In an edge implementation, the tool choice and the designer's optimization effort can yield very different results in terms of performance. For this reason, HW acceleration approaches and how they integrate into industry workflows have to be taken into account in the benchmark definition.

Consistent offloading of computation. Another practical challenge is to maximize the amount of offloaded computation to the accelerators: current libraries rely on the on-board CPUs for image decoding and decompression. While offloading these computations, using vendor accelerated libraries can be beneficial, it adds another dimension of complexity for benchmarking. The developed benchmarks should cover these and treat them consistently.

6 Concluding Thoughts on Impact

In this position paper, we presented the challenges associated with the deployment of ML inference for space applications and the need for the development benchmarks.We highlighted the appearance of FPGA-based accelerators and COTS solutions as a main trend in on-board data processing systems and highlighted the rising need for the development of ML inference benchmarks for space applications. These benchmarks enable systematic evaluation of the trade-offs of various approaches of inference accelerators for space applications and pave the path for future designs. Besides the direct impact on the space industry, such benchmarks can be adapted and applied to other embedded, resource-limited platforms with similar requirements.

[3] Similar designs can be done for the platforms of other vendors.

References

[1] Cornelius Dennehy. 2019. A NASA GN&C Viewpoint on On-Board Processing Challenges to Support Optical Navigation and Other GN&C Critical Functions. https://indico.esa.int/event/225/contributions/4249/

[2] David GONZALEZ-ARJONA, Al-varo JIMENEZ-PERALO, Paul BAJANARU, Arturo PEREZ, Alfonso RODRIGUEZ, Ruben DOMINGO, Antonio PASTOR, Miguel Angel VERDUGO, Andres OTERO, and Eduardo DE LA TORRE. 2019. HW Reconfigurable Processing Avionics for Space Vision-based Navigation. In *European Workshop on On-Board Data Processing (OBDP2019)*. https://indico.esa.int/event/225/contributions/4312/contribution.pdf

[3] E. W. Gretok, E. T. Kain, and A. D. George. 2019. Comparative Benchmarking Analysis of Next-Generation Space Processors. In *2019 IEEE Aerospace Conference*. 1–16. https://doi.org/10.1109/AERO.2019.8741914

[4] Vivek Kothari, Edgar Liberis, and Nicholas D. Lane. 2020. The Final Frontier: Deep Learning in Space. arXiv:2001.10362 [eess.SP]

[5] Maxim Neumann, Andre Susano Pinto, Xiaohua Zhai, and Neil Houlsby. 2019. In-domain representation learning for remote sensing. arXiv:1911.06721 [cs.CV]

[6] Olivier Notebaert and Alain Rossignol. 2019. On-Board Payload Data Processing requirements and technology trends. In *European Workshop on On-Board Data Processing (OBDP2019)*. https://indico.esa.int/event/225/contributions/4298/contribution.pdf

[7] Alessandro Pappalardo. [n.d.]. *Xilinx/brevitas*. https://doi.org/10.5281/zenodo.3333552

[8] Massimiliano Pastena. 2019. ESA Earth Observation on board data processing future needs and technologies. In *European Workshop on On-Board Data Processing (OBDP2019)*. https://indico.esa.int/event/225/contributions/3687/attachments/3357/4395/OBDP2019-S01-03-ESA_Pastena_ESA_Earth_Observation_On_board_data_processing_future_needs_and_technologies.pdf

[9] D. Petrick, A. Geist, D. Albaijes, M. Davis, P. Sparacino, G. Crum, R. Ripley, J. Boblitt, and T. Flatley. 2014. SpaceCube v2.0 space flight hybrid reconfigurable data processing system. In *2014 IEEE Aerospace Conference*. 1–20. https://doi.org/10.1109/AERO.2014.6836226

[10] V. J. Reddi, C. Cheng, D. Kanter, P. Mattson, G. Schmuelling, C. Wu, B. Anderson, M. Breughe, M. Charlebois, W. Chou, R. Chukka, C. Coleman, S. Davis, P. Deng, G. Diamos, J. Duke, D. Fick, J. S. Gardner, I. Hubara, S. Idgunji, T. B. Jablin, J. Jiao, T. S. John, P. Kanwar, D. Lee, J. Liao, A. Lokhmotov, F. Massa, P. Meng, P. Micikevicius, C. Osborne, G. Pekhimenko, A. T. R. Rajan, D. Sequeira, A. Sirasao, F. Sun, H. Tang, M. Thomson, F. Wei, E. Wu, L. Xu, K. Yamada, B. Yu, G. Yuan, A. Zhong, P. Zhang, and Y. Zhou. 2020. MLPerf Inference Benchmark. In *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*. 446–459. https://doi.org/10.1109/ISCA45697.2020.00045

[11] Gencer Sumbul, Marcela Charfuelan, Begum Demir, and Volker Markl. 2019. Bigearthnet: A Large-Scale Benchmark Archive for Remote Sensing Image Understanding. *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium* (Jul 2019). https://doi.org/10.1109/igarss.2019.8900532

[12] Yaman Umuroglu, Nicholas J. Fraser, Giulio Gambardella, Michaela Blott, Philip Leong, Magnus Jahre, and Kees Vissers. 2017. FINN. *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays* (Feb 2017). https://doi.org/10.1145/3020078.3021744

[13] Ussama Zahid, Giulio Gambardella, Nicholas J. Fraser, Michaela Blott, and Kees Vissers. 2020. FAT: Training Neural Networks for Reliable Inference Under Hardware Faults. arXiv:2011.05873 [cs.LG]

[14] Xiaoxiang Zhu, Jingliang Hu, Chunping Qiu, Yilei Shi, Hossein Bagheri, Jian Kang, Hao Li, Lichao Mou, Guicheng Zhang, Matthias Häberle, Shiyao Han, Yuansheng Hua, Rong Huang, Lloyd Hughes, Yao Sun, Michael Schmitt, and Yuanyuan Wang. 2018. So2Sat LCZ42. https://doi.org/10.14459/2018MP1454690