Reference Implementations for Machine Learning Application Benchmark

Andreas Koch[†*], Gabriel Dax[*], Michael Petry[†*], Harvey Gomez[†], Amir Raoofy[*], Urvij Saroliya[*], Max Ghiglione[‡], Gianluca Furano[‡], Martin Werner[*], Carsten Trinitis[*], Martin Langer[§]
[*]Technical University Munich, [†]Airbus Defence and Space GmbH, [‡]European Space Agency, [§]Orbital Oracle Technologies

***Abstract*—This paper presents reference implementations for a multitude of space applications from the Machine Learning Application Benchmark. Reference implementations include the respective model, its on-board hardware implementation, test scripts and final benchmarking results. In publishing these reference implementations, we make a significant contribution to the benchmark and provide more insight into the viability of on-board machine learning applications.**

***Index Terms*—Machine learning, neural networks, benchmark, FPGA, datasets, power consumption**

I. Introduction

Many applications such as maps, communication, navigation and much more depend on orbiting satellites. One limiting factor when it comes to satellites or spacecrafts is the remote operation. It motivates automating as much of the operation process as possible in order to reduce the manual effort required on-ground, while increasing performance and reliability. To this end, machine learning (ML) can be employed in multiple instances, such as for anomaly detection [1]. Apart from remote operation, time criticality in earth observation and communication establishes a reason for on-board deployment of ML algorithms in these fields.

For this purpose, the use of deployment of machine learning algorithms increases in space demonstration missions, especially when it comes to miniaturized satellites and large-scale industrial projects, such as ESA's phi-sat. This is caused by the rise of commercial off-the-shelf (COTS) products and the rapid development of hardware that is able to infer deep learning models on radiation-protected hardware. [2]

The training and inference of ML algorithms on such hardware does face several challenges though. Firstly, respective hardware needs to be able to meet the demand of computations for ML algorithms, as well as storage resources. Consequently, specialized processors for deep learning are utilized and typically an FPGA is programmed to accommodate them. In this work, the Xilinx Vitis AI Deep Learning Processor Unit (DPU) is chosen as the main hardware acceleration option. Aside from it, one of the neural networks presented will be deployed with the Xilinx FINN framework.

In hardware accelerated machine learning, important aspects to consider are the power consumption and throughput of the accelerator. Secondly, integration into the industry's processing chain in terms of interfaces and communication via standards such as PUS needs further exploration. Lastly, a comparatively small number of non-commercial, non-classified, and labeled datasets are available, which is a problem when deep neural networks are trained for different scenarios, such as wildfire detection. Therefore, sharing datasets in domain-specific disciplines, such as spaceflight, is of high importance and higher transparency is required. This includes requirements, implementation, hardware, software and finally the performance of different approaches. For this reason, the **Machine Learning Application Benchmark**[1] (MLAB) [3] funded by ESA was introduced with its core in on-board space applications. It aims to provide a set of requirements for measuring the performance of ML algorithms, guidelines and implementations, which are deployable to different hardware accelerators.

The rest of the paper is organized as follows. Section II introduces the details about the baseline implementations and section III presents corresponding results. Finally, section IV concludes the article.

II. Reference Implementation

In this section, we introduce a baseline implementation for a benchmark in the domain of deep learning in a space environment using on-board hardware. While it is challenging to provide a single requirement for all scenarios, such as classification and segmentation, requirements on each reference implementation are provided in the following. Due to its diversity and individual constraints, each reference scenario provides a baseline dataset and implementation. While this differs from case to case, all references are deployed to the FPGA board Xilinx Zynq Ultracsale+ MPSoc ZCU102. This target platform supports a wide range of hardware acceleration frameworks such as Vitis AI and FINN. All in all, each of the following reference benchmark scenarios provides a neural network trained on a specified publicly available dataset. All the models are trained and referenced with the frameworks TensorFlow and Vitis AI and deployed to the above mentioned FPGA board to ensure comparability between the scenarios. While table I gives an overview of the datasets which have

This activity has been funded by the ESA General Support Technology Programme within the project Machine Learning Application Benchmark (MLAB). Corresponding Author: Andreas Koch (Email: andreas.c.koch@airbus.com)

[1]https://github.com/mwernerds/mlab-benchmark

TABLE I: Datasets Overview

Dataset	Samples	Resolution	Classes
NASA Anomaly	$701,664$	25×1	2
RadioML	$2,555,904$	1024×2	24
Fire Detection	$11,347$	64×64	2
EuroSAT	$27,000$	$256 \times 256 \times 13$	10
Airbus Airplane	103	$2560 \times 2560 \times 3$	2
Airbus Ship Detection	$200,000$	$768 \times 768 \times 3$	2

been used, the following sections represent the different benchmark use-cases. Moreover, if not mentioned differently all power consumption metrics, throughput, and accuracy will be considered to evaluate and compare the scenario.

A. Anomaly Detection – Light

Failure detection is a central task of all spacecraft deceives and should be done as close as possible to the incident on board to have a faster reaction time. Due to bandwidth limitations and a small visiting timeframe of satellites, the industry works towards an automated anomaly detection and prediction system using machine learning algorithms. One main challenge is the number of telemetry data from the spacecraft. The NASA Anomaly dataset provides anonymized real telemetry of the Soil Moisture Active Passive Satellite (SMAP), as well as from the Mars Science Laboratory (MSL) on the Curiosity rover. In contrast to solving this task with an LSTM model as in [1], the baseline model is chosen to be an NBEATS model [4]. It is a deep neural network with residual connections, specifically designed for time series forecasting. Since it solely utilizes different versions of feed-forward layers, we were able to deploy it to the DPU with only minor changes.

B. Radio Classification

The task of radio classification aims to archieve a high capacity on a dynamically shared channel. Neural networks can be used to detect the type of signal and the modulation which is on a specific channel. The Open RadioML Synthetic Benchmark dataset is used as a baseline for this scenario. The ResNet model in [5] with four residual layers connected to a softmax output is used as a baseline. Furthermore, according to [6] a vector representation for the IQ plane supports the stimulation of deeper layers.

C. Image Classification – Multispectral

Multispectral images sensed from space need to be processed in order to detect their content. Detection can be done on a per pixel basis or by tiling the image into smaller patches for classification. For wildfire detection, tiles are categorized into fire, no fire, and smoke. This reference implementation uses a labeled dataset, created from 11,347 Sentinel-2 images. For the model, a VGG16 convolutional neural network was used. It creates bounding boxes for the full image for positive detections.

D. Image Classification – Heavy

Convolutional neural networks (CNNs) all come in different forms and depths. Some CNNs such as MobileNet are build to run in edge devices, while others have a higher complexity and depth. An example for the second category would be DenseNet. Due to its increased complexity, it has the ability to reach high accuracies, although this comes at a cost of high resource utilization. This scenario of "heavy" image covers the use of deeper networks and the DenseNet161 is chosen for the reference implementation. The EuroSAT dataset [7] serves as the baseline, while only RBG channels are considered.

E. Image Classification – Light

Contrary to the last scenario, here we investigate the performance of a smaller convolutional neural network. For this purpose, the MobileNet model was chosen, combined with the Airbus ship detection dataset. This model was deployed using the FINN framework.

F. Image Object Detection

This scenario is about object detection and the aim is to produce bounding boxes around detected objects, in line with [2]. A YOLO model has been selected as the reference model and accuracy will be measured via average precision (AP). The subject matter is the Airbus Aircraft dataset.

G. Image Segmentation

Compared to scenario II-E, this scenario segments the image on a per pixel basis instead of providing bounding boxes. Similarly, the Airbus Ship detection dataset is used as a reference dataset, while a UNet based model called ResNet50 was chosen as the reference model. The performance metric is intersection over union for a specified confidence threshold.

III. Results

All implementations use quantization, which generally degrades accuracy while improving throughput and energy efficiency. Quantization entails reducing precision to 8-bit integers or 16-bit float types. Although, adjusting the range and resolution of quantization can recover the loss to a certain degree. To this end, a calibration dataset is required to optimize the quantization for a specific use case. Consequently, we will publish calibration datasets for every reference implementation.
Taking a look at Table II, the first thing that is apparent is the significantly lower model size for scenarios A and B. Both scenarios operate with time series data, which require a much smaller model complexity to achieve good performance, in comparison to neural networks processing images. Naturally, both show a higher throughput in terms of frames per second. Accordingly, also the difference between idle and maximum power is generally lower for smaller neural networks. The on-board performance metrics refer to the respective performance of the quantized model. For scenario A specifically, one model was trained on every file available of the NASA Anomaly dataset and the evaluation is in line with the evaluation method of [1]. Predicted and real anomaly sequences are

TABLE II: Reference implementations details and on-board performance

Use Case	Model Size [Parameters]	Framework	OS	PL	PS	PL resources	On-Board Perf. (Metric)	Throughput [Frames/s]	Idle [W]	Max [W]	Energy [µJ/Bit]
A	140k	Vitis AI	Petalinux	1 DPU	1 Thread	30%	67.4% (F0.5)	1634.25	4.9	5.1	0.147
B	179k	Vitis AI	Petalinux	1 DPU	1 Thread	30%	58.0% (Acc.)	556.98	3.8	4.3	0.116
C	15M	Vitis AI	Petalinux	1 DPU	5 Threads	30%	92.0% (Acc.)	438.5	-	-	-
D	12.6M	Vitis AI	Petalinux	3 DPUs	4 Threads	90%	89.1% (Acc.)	127.77	6.0	14.09	0.013
E	3.4M	FINN	PYNQ OS	Custom IP	-	90%	96.0% (Acc.)	25.74	3.2	4.2	-
F	37.9M	Vitis AI	Petalinux	3 DPUs	1 Thread	90%	87.0% (AP)	10.4	8.3	11.7	0.536
G	31.4M	Vitis AI	Petalinux	1 DPU	1 Thread	30%	61.0% (IoU)	9.0	3.8	7.8	0.061

compared and a true positive is counted in the case of an overlap. After aggregating true positives, false positives and false negatives over all files, the F0.5-score is calculated to be 67.4%. The accuracy of 58.0% of scenario B was averaged across 1000 sampled signal-to-noise ratios of the range -20dB to 30 dB. Moving on to the scenarios involved with image processing, a model with 15 million parameters was trained on the fire detection dataset and achieved an accuracy of 92%, as well as a throughput of 438.5 FPS for a frame of size 64×64. The power and energy measurement is still missing for scenario C and will be added with the submission of the reference implementation at a later point. Considering the on-board implementation for scenario D, it is clear that this implementation achieves the lowest energy cost for the per Bit computation. A reason could be the deployment of 3 DPUs, in combination with running 4 threads in parallel. As a consequence, implementation D shows the highest peak power and the widest margin of peak to idle power. Scenario E was the sole scenario utilizing the Xilinx FINN framework for deployment. For this scenario, the Airbus Ship Detection Kaggle dataset was treated as a binary classification problem: an image can either contain a ship or not. Compared to image segmentation, this task is less complex and thus an overall accuracy of 96% could be reached. Regarding airplane detection in scenario F, a YOLOv4 model was quantized and compiled along the guidelines outlined in the Vitis AI tutorials and an overall average precision of 87% was attained on-board. The on-board implementation for the segmentation model of scenario G reaches an accuracy in terms of intersection-over-union of 61% with a confidence threshold of 0.1. As only one threshold was tested on-board, a full comparison to the Kaggle leaderboard[2] is not viable as of this point.

IV. Conclusion

In this paper, we presented a multitude of reference implementations for ML space applications. Covering anomaly detection for satellite telemetry, telecommunication and earth observation applications, these implementations introduce a baseline performance for every application in the MLAB benchmark. In publishing these results, we hope to make a significant contribution to the benchmark and provide more insight into the viability of on-board machine learning applications.

[2]https://www.kaggle.com/competitions/airbus-ship-detection/leaderboard

Acknowledgment

This project has been funded by the ESA General Support Technology Programme.

References

[1] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 387–395. [Online]. Available: https://doi.org/10.1145/3219819.3219845

[2] A. Raoofy, G. Dax, V. Serra, M. Ghiglione, M. Werner, and C. Trinitis, "Benchmarking and feasibility aspects of machine learning in space systems," in *Proceedings of the 19th ACM International Conference on Computing Frontiers*, ser. CF '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 225–226. [Online]. Available: https://doi.org/10.1145/3528416.3530986

[3] A. Koch, M. Petry, M. Ghiglione, A. Raoofy, G. Dax, G. Furano, M. Werner, C. Trinitis, and M. Langer, "Machine learning application benchmark," in *Proceedings of the 20th ACM International Conference on Computing Frontiers*, ser. CF '23. New York, NY, USA: Association for Computing Machinery, Aug 2023, p. 229–235. [Online]. Available: https://dl.acm.org/doi/10.1145/3587135.3592769

[4] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-beats: Neural basis expansion analysis for interpretable time series forecasting," Mar 2022. [Online]. Available: https://openreview.net/forum?id=r1ecqn4YwB

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Dec 2015, arXiv:1512.03385 [cs]. [Online]. Available: http://arxiv.org/abs/1512.03385

[6] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, 2018.

[7] P. Helber, B. Bischke, A. Dengel, and D. Borth, "Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 7, pp. 2217–2226, 2019.