

# Reference Implementations for Machine Learning Application Benchmark

Andreas Koch<sup>†\*</sup>, Gabriel Dax<sup>\*</sup>, Michael Petry<sup>†\*</sup>, Harvey Gomez<sup>†</sup>, Amir Raoofy<sup>\*</sup>, Urvij Saroliya<sup>\*</sup>,  
Max Ghiglione<sup>‡</sup>, Gianluca Furano<sup>‡</sup>, Martin Werner<sup>\*</sup>, Carsten Trinitis<sup>\*</sup>, Martin Langer<sup>§</sup>

<sup>\*</sup>Technical University Munich, <sup>†</sup>Airbus Defence and Space GmbH, <sup>‡</sup>European Space Agency, <sup>§</sup>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. [?]

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 in this work and typically an FPGA is programmed to accommodate them. In this work, multiple such processors will be employed, namely the Xilinx Vitis AI Deep Learning Processor Unit (DPU) and the Deep Learning Processor (DLP) provided by the Matlab Deep Learning HDL toolbox. Thereby, 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** (MLAB) 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 Ultrascale+ MPSoc ZCU102. This target platform supports a wide range of hardware acceleration frameworks such as VitisAI, FINN, and MATLAB’s Deep Learning HDL toolbox. 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 VitisAI 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 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.

TABLE I: Datasets Overview

Dataset	Samples	Resolution	Classes
RadioML	2,555,904	1024 × 2	24
EuroSAT	27,000	256 × 256 × 13	10
Airbus Ship Detection	200,000	768 × 768 × 3	2

#### A. Anomaly Detection – Light

Failure detection is a central task of all spacecraft deceives and must be done as close as possible to the incident on board. 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, which contains a large volume of multivariant labeled telemetry data from different devices (MSL, SMAP), is selected as a reference. Same as in [1], the baseline model is chosen to be a LSTM model. It will be deployed via the MATLAB Deep Learning HDL toolbox.

#### B. Radio Classification

The task of radio classification aims to archive 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 [2] with four residual layers connected to a softmax output is used as a baseline. Furthermore, according to [3] 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 [4] serves as the baseline, while only RGB 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 purposee, 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 [5]. 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

This section presents results for multiple reference implementations available in Table II. 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.

Considering the on-board implementation for scenario II-D, it is clear that this implementation achieves the highest throughput and the lowest energy cost for the per Bit computation. A reason could be the deployment of 3 Xilinx Vitis AI Deep Learning Processing Units (DPUs). Accordingly, multiple threads process the data to supply the different DPUs. As a consequence, implementation II-D shows the highest peak power and the widest margin of peak to idle power. The on-board implementation for the segmentation model of scenario II-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<sup>1</sup> is not viable as of this point. A higher throughput and energy cost per Bit can be explained by an overall larger model size compared to implementation II-D and the use of only one DPU. Surprisingly, the reference implementation II-B exhibits the lowest throughput and highest energy cost so far, despite having the smallest model. The size of the model is likely also the reason for the low margin of peak to idle power.

<sup>1</sup><https://www.kaggle.com/competitions/airbus-ship-detection/leaderboard>

TABLE II: Reference implementations details and on-board performance

Use Case	Model Size [Parameters]	Framework	OS	PL	PS	PL resources	On-Board Accuracy	Throughput [MB/s]	Idle [W]	Max [W]	Energy [ $\mu$ J/Bit]
B	179k	Vitis AI	Petalinux	1 DPU	1 Thread	30%	56.0%	4.563	3.8	4.3	0.116
D	12.6M	Vitis AI	Petalinux	3 DPUs	4 Threads	90%	89.1%	96.550	6.0	14.09	0.013
F	NaN	Vitis AI	Petalinux	3 DPUs	1 Thread	90%	87.0% (AP)	2.726	8.3	11.7	0.536
G	31.4M	Vitis AI	Petalinux	1 DPU	1 Thread	30%	61.0% (IoU)	5.419	3.8	7.8	0.061

#### IV. CONCLUSION

In this paper, we presented a multitude of reference implementations for ML space applications. Covering anomaly detection for satellite telemetry 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.

#### 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] 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>
- [3] 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.
- [4] 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.
- [5] 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>