ASCEND
23-25 October 2023, Las Vegas, Nevada
ASCEND 2023

10.2514/6.2023-4718

Deploying Artificial Intelligence Capabilities by Hybridizing a Neural Network on a Satellite

L. Chavier[1], B. Bonham-Carter[2], H. Burd[3], T. Heydrich[4], G. O'Shea[5], J. Prud'homme[6], N. Ayyappan[7], M. Maharib[8], T. Ganesalingam[9], A. Higginson[10], E. Smal[11], S. Pillay[12], K. Raimalwala[13], Y. Brown[14], A. J. Macdonald[15], M. Faragalli[16]
Mission Control, Ottawa, ON, K1R 6N5, Canada

Future space missions require more autonomous capabilities. Modern autonomy-enabling software, such as deep learning algorithmic approaches, offers the potential to enhance current and future missions but only if these algorithms can be matched to the constraints and requirements of flight hardware, including limited size, weight, power, memory, and radiation tolerance. An important option to accelerate the deployment of deep learning onboard is the hybridization across software and firmware in embedded flight computers. In this paper we present one of the first cases of hybridization of a convolutional neural network in space, validated onboard ESA's OPS-SAT with the SmartCam convolutional neural network. We use a custom compiler and runtime designed to deploy neural networks across different hardware targets for space missions and implement a generic matrix multiplier to execute multiply-accumulate cycles from the SmartCam architecture on the onboard field programmable gate array (FPGA). We demonstrate identical output classification performance between the original and hybridized implementations of SmartCam with one hundred images taken in orbit.

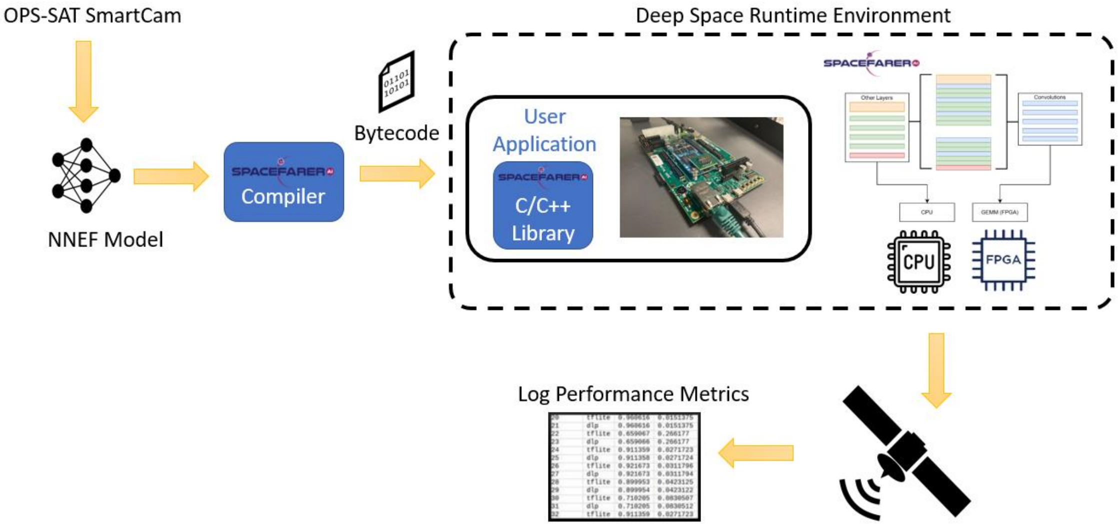
I. Introduction

As humanity ventures further into deep space, not just to explore but to inhabit, our space systems will require more autonomous capabilities in and beyond earth orbit. The Global Exploration Roadmap (GER) [1], [2] describes

[1] Principal Software Developer
[2] Embedded Software Engineer
[3] Software Engineer
[4] Intermediate Software Engineer
[5] Junior AI Specialist
[6] Junior Software Engineer
[7] Robotics Co-op Student
[8] Robotics Co-op Student
[9] Robotics Co-op Student
[10] Principal Software Engineer
[11] Principal Software Engineer
[12] Senior AI Specialist.
[13] Senior Manager, Business Development
[14] Director of People & Programs
[15] Senior AI Specialist, AIAA Young Professional
[16] Chief Technology Officer

the joint plans of dozens of space agencies across the globe to create the space exploration and infrastructure necessary to explore, live, and work in deep space. To accomplish these plans requires greater autonomous capabilities for remote sensing, planetary science, station keeping, and other activities as captured by the GER Critical Technology Needs [3] such as GER-020 Robots Working Side-by-Side with Suited Crew or GER-021 Autonomous Vehicle Management Systems. Machine learning techniques, including deep learning and reinforcement learning, are currently the state-of-the-art artificial intelligence technologies for enabling these autonomy capabilities in spacecraft systems. The Machine Learning Technology Readiness Levels (ML-TRL) [4] describe unique challenges associated with stewarding machine learning based technologies, including management and maintenance of data, models, and software. In spacecraft flight systems, size, weight, radiation, bandwidth, and power constraints create an additional unique set of requirements [5]. Examples of successful AI for enabling spacecraft autonomy include the AutoNav, OnBoard Planner, and AEGIS system on NASA’s Perseverance [6], ESA’s OPS-SAT Earth observation SmartCam model [7], [8], ESA’s Φ-sat-1 hyperspectral cloud segmentation model [9], Palantir’s onboard ship detection and segmentation models [10], Mission Control’s own MoonNet model [11], and the WorldFloods model deployed by Trillium Technologies and ESA Φ-Lab as an ML payload on the D-Orbit Wild Ride mission [12]. With additional onboard AI planned, such as onboard the International Space Station [13] or ESA’s CHIME mission [14], there is a need for platform-agnostic software tools that can robustly deploy neural network deep learning models on spaceflight computers. To that end, Mission Control has developed a flight deployment software tool, the Spacefarer AI Deployment Toolkit, and tested it on multiple operational space platforms.

In this paper we present Mission Control’s flight deployment software for commissioning and running deep learning models to add new autonomy capabilities to spacecraft, a crucial component of our larger deep learning pipeline [11]. Our model for deployment uses a custom multi-stage neural network compiler and runtime environment that supports the Khronos Group Neural Network Exchange Format (NNEF) format [15] to deploy lightweight, robust convolutional neural networks (CNN) onboard systems-on-a-chip (SoC) on embedded flight computers. The need to build our own neural network deployment tools specifically for spaceflight was driven by two different flight technology demonstrations, the deployment of the MoonNet CNN onboard the ispace HAKUTO-R mission that launched December 11th, 2022 with an onboard Xilinx Zynq 7020 system-on-a-chip (SoC) and the subject of this paper: our flight demonstration on ESA’s OPS-SAT in low-earth orbit, which was performed in August of 2023 using an Intel Cyclone V onboard SoC. Intel and Xilinx have their own, manufacturer-specific deployment tools, OpenVINO and VitisAI respectively, however a trade-off analysis demonstrated that building a hardware agnostic software tool specific to the needs of space applications, which we call the Spacefarer AI Deployment Toolkit, would reduce the complexity of supporting multiple neural networks deployments across multiple flight demonstrations and hardware targets. We showcase the effectiveness of this approach by describing the path to validate and deploy a hybridized Spacefarer AI version of the SmartCam image classification CNN [7] across the CPU and FGPA of the OPS-SAT Satellite Experimental Processing Platform (SEPP) and compare the results to a pure Spacefarer AI CPU implementation. As noted by Furano *et al.* [5], space data processing systems often employ FPGAs for algorithmically intensive processing because compared to terrestrial commercial-off-the-shelf (COTS) components like CPUs, Graphics Processing Units (GPUs), or Vision Processing Units (VPUs) they offer better radiation tolerance and can significantly increase the performance per watt. Previously, Abderrahmane *et al.* [16] demonstrated the use of Spiking Neural Networks (SNNs) onboard OPS-SATs FPGA while Lemaire *et al.* [17] tested spiking, fully connected, and convolution architectures. To the author’s knowledge this makes the current work one of the first times a hybridized CNN has been flown in low earth orbit. Figure 1 shows the essential workflow elements of the experiment to hybridize SmartCam and test the CPU-only implementation against the CPU & FPGA implementation in orbit.



```
20    tflite  0.960616  0.0151375
21    dlp     0.960616  0.0151375
22    tflite  0.659067  0.266177
23    dlp     0.659066  0.266177
24    tflite  0.911359  0.0271723
25    dlp     0.911358  0.0271724
26    tflite  0.921673  0.0311796
27    dlp     0.921673  0.0311794
28    tflite  0.899953  0.0423125
29    dlp     0.899954  0.0423122
30    tflite  0.710205  0.0830507
31    dlp     0.710205  0.0830512
32    tflite  0.911359  0.0271723
```

Figure 1: Set up of the deployed OPS-SAT experiment, which compared runtime performance between a pure Spacefarer AI CPU implementation of the SmartCam CNN to a hybridized implementation that used the CPU and FPGA of the Cyclone V SoC.

OPS-SAT is a 3U nanosatellite owned and operated by ESA, with the purpose of demonstrating technological firsts in-orbit, including in AI and autonomy for mission operations. Once launched in December 2019, OPS-SAT could perform experiments proposed by ESA members, including academia and industry entities within member countries. OPS-SAT uses a convolutional neural network called SmartCam to ingest data from a three-band, optical sensor onboard and classify incoming imagery as “Earth”, “Edge of Space”, or “Bad” to give an indication whether an image should be prioritized for downlink or not, forming the first use of AI by ESA for onboard scheduling on a mission in flight [7], [8]. The experiment to develop and hybridize SmartCam occurred from 2022-2023, cumulating in a flight demonstration in August 2023.

II. Experimental Methodology

FPGAs offer a balance of reconfigurability, generalizability, and utilization efficiency but FPGA deep learning frameworks are still in their infancy [18]. Our hybridized neural network FPGA implementation advances how AI, and in particular CNNs, can be deployed on spacecraft using NNEF, an open and standard data format for exchanging information about trained neural networks. To develop the experiment, we undertook a series of steps on a MitySOM development board, which also uses the Cyclone V SoC, to characterize the SmartCam model operations and design a Generic Matrix Multiplier (GEMM) algorithm that can effectively hybridize the SmartCam model. The first analysis step consisted of evaluating two aspects: (1) how the number of MACs (multiply-accumulate cycles) is distributed across the layers on the network, and (2) the memory requirements for the operands and results of each layer. The computational requirements of networks are usually measured in MACs as they are the basic arithmetic operation which composes a convolutional layer. Convolutions usually account for 90% or more of the MACs on CNNs, and such MACs occur by repeatedly accessing the same elements on input feature maps and filters. This means that a significant number of arithmetic operations occur across the same elements. FPGAs can perform many arithmetic operations in parallel, but to do this efficiently, the data must generally be in on-chip memory inside the FPGA. This memory is typically limited in size, and indeed is only given by at most 553 M10K blocks for the Cyclone V on OPS-SAT. Therefore, we concluded that the operations that entail performing a high number of arithmetic operations on top of limited amounts of memory, such as convolutions, are the ones with the highest potential for performance gains. This analysis resulted in limiting the FPGA-accelerated network operations to convolutional layers.

Figure 2 depicts the hybridization of the SmartCam model on a high level, showing which layers are being sent to the GEMM and which are run on the CPU. We report the results of stepwise performance testing of this implementation on development boards and Engineering Models, comparing the accuracy, precision, and average elapsed time and CPU time of the pure CPU implementation to the hybridized CPU and FPGA approach, using a dataset of one hundred OPS-SAT images labelled as ‘Earth’, ‘Edge’, or ‘Bad’ by the SmartCam model. We also report results for the flight experiment onboard the spacecraft, to our knowledge one of the first implementations of a hybridized convolutional neural network implemented on a spacecraft, which additionally relied on orbital planning and capture steps to acquire and process new images in near-real time.

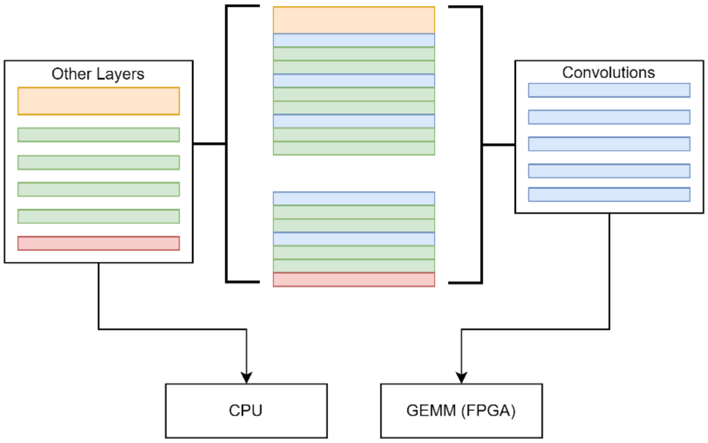


Figure 2: Model layer CPU/FPGA selection demonstrating how the convolution operations of SmartCam are identified and run on the FPGA via the GEMM, rather than the CPU.

The general information flow and decision-making of the GEMM is given in Figure 3, showing the logical flow for two matrices A and B. Due to the limited on-chip size only one matrix is loaded into memory while the other is being streamed in, column by column, to perform the matrix multiplication. The output column is then streamed back to external memory.

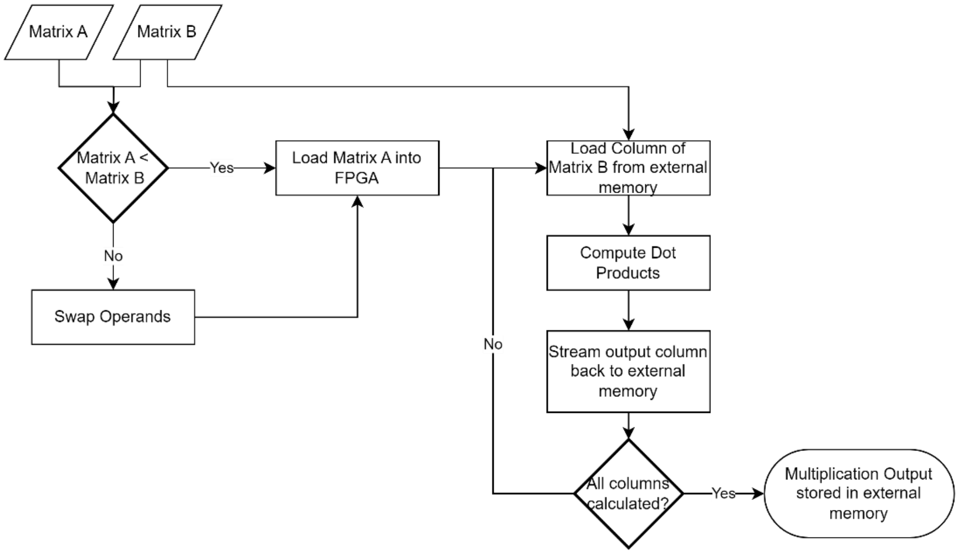

Matrix A
Matrix B
Matrix A < Matrix B
Yes
Load Matrix A into FPGA
Load Column of Matrix B from external memory
No
Swap Operands
Compute Dot Products
No
Stream output column back to external memory
All columns calculated?
Yes
Multiplication Output stored in external memory

Figure 3: Matrix multiplication software flow.

A stylized view of the matrix multiplication process from the FPGA's perspective is shown in Figure 4. The rows of Matrix A and columns of Matrix B are combined to create a stream of output columns that comprise the output matrix. Performing this process with many of the core matrix multiplications across the SmartCam model, which uses the MobileNetV2 backbone [19], the hybridized scheme is executed with part of the neural network operations running on the Cyclone V CPU and part on the FPGA.

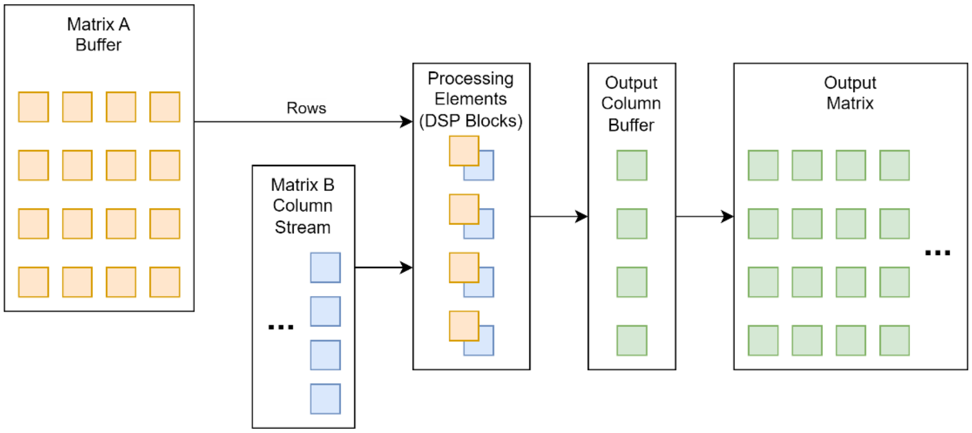


Figure 4: GEMM data flow.

III. Results & Discussion

In this section we present the sequence of steps that led to the in-flight run of the experiment, and the obtained results. As noted by Lavin *et al.* [4] and Mateo-Garcia *et al*. [12], machine learning payloads for spaceflight are subject to additional requirements to ensure robustness and reliability, due to their complex reliance on both data and software. Validation and verification processes for machine learning payloads are still being established and this paper highlights the methodology we successfully used to deploy our hybridized machine learning payload to a flight demonstration. It is important to note that this process was executed while OPS-SAT was already in orbit, making it distinct from a process developed from pre-flight mission requirements. Four stages were executed sequentially, with increasing levels of complexity, as illustrated in Figure 5. During experiment development, any changes triggered the creation of a new release and a return to the initial stage for validation across the whole chain. The stages can be described as follows:

- **Breadboard Model stage:** the experiment was initially run using a development board based on the MitySOM system-on-module. A test dataset was used at this stage given the lack of an integrated sensor to provide relevant images.
- **First Engineering Model stage:** once validated on the breadboard model, the experiment was then run on the flatsat provided by ESA using the same test dataset.
- **Second Engineering Model stage:** after validation with the test dataset, the experiment was run on ESA's flatsat using the flatsat camera. While the images were not relevant in terms of model performance evaluation, this extra stage allowed validation of the interface with the satellite's camera.
- **Flight Model stage:** at the final stage, the experiment was run on the satellite using data captured from the camera in flight.

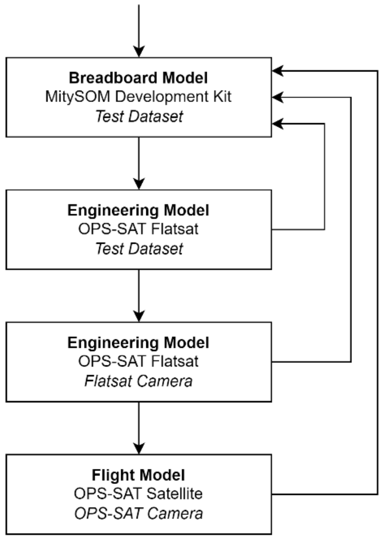


Figure 5: Stages leading to in-flight experiment.

A. Test Dataset

To validate the experiment software across the multiple stages leading to the in-flight experiment, a test dataset was created using images previously captured by OPS-SAT and made available by ESA. A set of 100 random images was sampled from the labeled thumbnails provided by ESA and used to compare the results obtained by different implementations of the SmartCam model. Test runs using the test dataset were successful, obtaining similar results compared to what was obtained during the in-flight experiment (see Model Performance section).

B. Spacefarer AI CPU and FPGA Use

Using the test dataset, two different versions of the experiment were run on ESA's engineering model of OPS-SAT (flatsat): one using only the CPU for the convolution layers and one hybrid version using the FPGA to compute some of the convolution layers. The results in Table 1 show that an amount of CPU offloading was achieved in the hybrid version versus the CPU-only implementation. For a similar elapsed inference time averaged over all the 100 images, the average CPU time measured by the operating system was lower in the hybrid version compared to the CPU-only version of the experiment. When using the hybrid version, that represents additional computational capability available for other processes that might be running on the CPU.

Table 1: CPU and FPGA use measurements.

Implementation	Average Elapsed Time (ms)	Average CPU Time (ms)
Spacefarer AI CPU	8644.13	8260.88
Spacefarer AI Hybrid (CPU+FPGA)	8625.95	8034.41

C. Model Performance

After validating the FPGA operation and confirming CPU offloading on the engineering model, the in-flight experiment was run in August 2023. A total of 100 images were captured and processed in orbit by both the hybrid implementation and a SmartCam implementation using TensorFlow Lite equivalent to the one originally used by ESA [7]. The goal was to verify the model outputs in both implementations, and the results show that the hybrid implementation using the FPGA yielded the same output classification results as the TensorFlow Lite implementation. As seen in Table 2, both implementations predicted the exact same classes, with a minimal deviation in the average negative log likelihood (NLL) loss computed for both models over 100 images. Figure 6 shows some of the images captured during the in-flight experiment.

Table 2: Model output comparison between hybrid and TensorFlow Lite implementations.

Implementation	Accuracy	Precision	Sensitivity	Specificity	F1 Score	Avg. Loss (NLL)
TensorFlow Lite	98%	0.98	0.55	0.78	0.99	0.201807
Hybrid (CPU+FPGA)	98%	0.98	0.55	0.78	0.99	0.201811

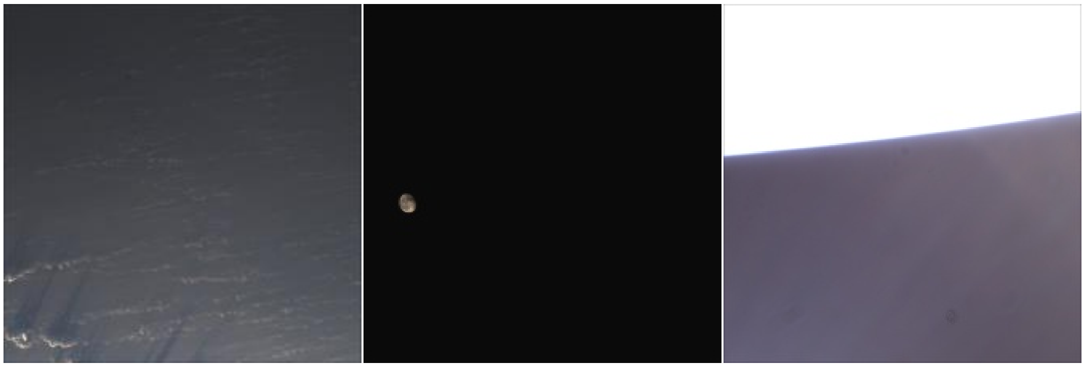


Figure 6: Images acquired by ESA's satellite OPS-SAT during Mission Control's experiment and processed by the Spacefarer AI hybridized SmartCam onboard the satellite. Output classification labels between ESA's original SmartCam and the hybridized SmartCam agree across all images acquired.

IV. Conclusion

Spaceflight missions provide an ideal use case for enhanced autonomy given their distance from earth, their tendency to explore new and unknown environments, and the comparative difficulty of sending humans outside of earth's atmosphere. A challenge to enabling greater autonomy on spaceflight computers is the unique and often harsh environments in which they operate, their limited memory and power compared to compute available terrestrially, and the need to use software tooling specific to the embedded target hardware and mission. In this work we detail the steps to validate and fly a hybridized neural network using the Spacefarer AI Deployment Toolkit, which was designed with a hardware-agnostic intent to support embedded targets from multiple manufacturers and has now been used to deploy neural networks into low earth orbit and for a lunar demonstration. Within the context of the Deployment Toolkit, we demonstrate the potential for hybridization to free up the CPU of an SoC by moving MAC operations to the FPGA firmware. We believe hybridization is a potential tool to address onboard compute challenges and unlock autonomous capabilities. The hybridization of SmartCam onboard OPS-SAT provides a baseline that can be combined with other techniques such as onboard training [20], knowledge distillation [21], and different learning models [22].

Acknowledgments

The authors acknowledge support from ESA's Open Space Innovation Platform (OSIP) and wish to thank the entire OPS-SAT operations team for their professionalism and support throughout. In particular, the authors would like to thank Georges Labrèche, Dave Evans, Léa Dubreil, and Rodrigo Laurinovičs.

References

[1] 'ISECG Global Exploration Roadmap', ISECG (International Space Exploration and Coordination Group), Jan. 2018. Accessed: Aug. 17, 2018. [Online]. Available: https://www.globalspaceexploration.org/wordpress/wp-content/uploads/2013/10/SKGs-Summary-Table-Final-for-Posting.pdf

[2] 'Global Exploration Roadmap Supplement October 2022 - Lunar Surface Exploration Scenario Update'. International Space Exploration Coordination Group (ISECG), 2022. [Online]. Available: https://www.globalspaceexploration.org/wp-content/isecg/GER_Supplement_Update_2022.pdf

[3] 'Global Exploration Roadmap Critical Technology Needs'. International Space Exploration Coordination Group (ISECG), 2019. [Online]. Available: https://www.globalspaceexploration.org/wp-content/uploads/2019/12/2019_GER_Technologies_Portfolio_ver.IR-2019.12.13.pdf

[4] A. Lavin *et al.*, 'Technology readiness levels for machine learning systems', *Nat. Commun.*, vol. 13, Art. no. 1, 2022, doi: 10.1038/s41467-022-33128-9.

[5] G. Furano *et al.*, 'Towards the Use of Artificial Intelligence on the Edge in Space Systems: Challenges and Opportunities', *IEEE Aerosp. Electron. Syst. Mag.*, vol. 35, no. 12, pp. 44–56, Dec. 2020, doi: 10.1109/MAES.2020.3008468.

[6] V. Verma *et al.*, 'Autonomous robotics is driving Perseverance rover's progress on Mars', *Sci. Robot.*, vol. 8, no. 80, p. eadi3099, Jul. 2023, doi: 10.1126/scirobotics.adi3099.

[7] G. Labreche *et al.*, 'OPS-SAT Spacecraft Autonomy with TensorFlow Lite, Unsupervised Learning, and Online Machine Learning', in *2022 IEEE Aerospace Conference (AERO)*, Big Sky, MT, USA: IEEE, Mar. 2022, pp. 1–17. doi: 10.1109/AERO53065.2022.9843402.

[8] G. Labrèche, D. Evans, D. Marszk, T. Mladenov, V. Shiradhonkar, and V. Zelenevskiy, 'Artificial Intelligence for Autonomous Planning and Scheduling of Image Acquisition with the SmartCam App On-Board the OPS-SAT Spacecraft', *AIAA Scitech 2022 Forum*, Jan. 2022, doi: https://doi.org/10.2514/6.2022-2508.

[9] G. Giuffrida *et al.*, 'CloudScout: A Deep Neural Network for On-Board Cloud Detection on Hyperspectral Images', *Remote Sens.*, vol. 12, no. 14, p. 2205, Jul. 2020, doi: 10.3390/rs12142205.

[10] R. Imig and M. Rehman, 'Palantir Edge AI in Space'. [Online]. Available: https://blog.palantir.com/edge-ai-in-space-93d793433a1e

[11] A. J. Macdonald *et al.*, 'Enabling Autonomy with a Deep Learning Framework for Planetary Exploration', in *ASCEND 2022*, American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2022-4295.

[12] G. Mateo-Garcia *et al.*, 'In-orbit demonstration of a re-trainable machine learning payload for processing optical imagery', *Sci. Rep.*, vol. 13, no. 1, p. 10391, Jun. 2023, doi: 10.1038/s41598-023-34436-w.

[13] 'OrbitalAI IMAGIN-e', AI4EO Challenges by ESA Φ Lab. [Online]. Available: https://platform.ai4eo.eu/orbitalai-imagin-e

[14] R. Vitulli *et al.*, 'CHIME: THE FIRST AI-POWERED ESA OPERATIONAL MISSION', 2022.

[15] 'Neural Network Exchange Format (NNEF)', Khronos Group. [Online]. Available: https://www.khronos.org/nnef#:~:text=A%20stable%2C%20flexible%20and%20extensible%20standard%20that%20equipment,and%20the%20inference%20engine%20used%20to%20execute%20it.

[16] N. Abderrahmane, B. Miramond, E. Kervennic, and A. Girard, 'SPLEAT: SPiking Low-power Event-based ArchiTecture for in-orbit processing of satellite imagery', in *2022 International Joint Conference on Neural Networks (IJCNN)*, Padua, Italy: IEEE, Jul. 2022, pp. 1–10. doi: 10.1109/IJCNN55064.2022.9892277.

[17] E. Lemaire *et al.*, 'An FPGA-Based Hybrid Neural Network Accelerator for Embedded Satellite Image Classification', in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, Seville, Spain: IEEE, Oct. 2020, pp. 1–5. doi: doi: 10.1109/ISCAS45731.2020.9180625.

[18] S. Mittal, 'A survey of FPGA-based accelerators for convolutional neural networks', *Neural Comput. Appl.*, vol. 32, no. 4, pp. 1109–1139, Feb. 2020, doi: 10.1007/s00521-018-3761-1.

[19] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, 'MobileNetV2: Inverted Residuals and Linear Bottlenecks', presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520. Accessed: Jul. 08, 2020. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2018/html/Sandler_MobileNetV2_Inverted_Residuals_CVPR_2018_paper.html

[20] V. Růžička *et al.*, 'Fast model inference and training on-board of Satellites'. arXiv, Jul. 17, 2023. Accessed: Jul. 31, 2023. [Online]. Available: http://arxiv.org/abs/2307.08700

[21] P. Bosowski, N. Longépé, B. Le Saux, and J. Nalepa, 'Knowledge Distillation for Memory-Efficient On-Board Image Classification of Mars Imagery', in *IEEE International Symposium on Geosciene and Remote Sensing (IGARSS)*, Pasadena, CA: IEEE, Jul. 2023. [Online]. Available: https://2023.ieeeigarss.org/view_paper.php?PaperNum=3974

[22] S. Pillay *et al.*, 'Federated Learning on Edge Devices in a Lunar Analogue Environment', in *IEEE International Symposium on Geosciene and Remote Sensing (IGARSS)*, Pasadena, CA, Jul. 2023. [Online]. Available: https://2023.ieeeigarss.org/view_paper.php?PaperNum=2750