The ISOLDE Space Demonstrator: A Platform for AI Applications on Satellites

Emanuele Valpreda[1], Roberto Morelli[1], Antonio Sciarappa[1], Davide Di Ienno[2], Carlo Ciancarelli[2], Paolo Serri[2], Valeria Parascandolo[2], Antonio Leboffe[2], Dario Pascucci[2], Daniele Gregori[3], Mattia Paladino[3], Daniele Jahier Pagliari[4], Alessio Burrello[4], Sara Vinco[4], Gianvito Urgese[4], Maurizio Martina[4], Guido Masera[4], William Fornaciari[5], Federico Reghenzani[5], Andrea Galimberti[5], Davide Zoni[5], Giovanni Agosta[5], Andrea Acquaviva[6], Francesco Conti[6]

[1]*Leonardo SpA, Italy*, [2]*Thales Alenia Space Italia S.p.A., Italy*, [3]*E4 Computer Engineering SpA, Italy*, [4]*Politecnico di Torino, Torino, Italy*, [5]*Politecnico di Milano, Milano, Italy*, [6]*University of Bologna, Bologna, Italy*

Email:{emanuele.valpreda, roberto.morelli, antonio.sciarappa}@leonardo.com,
{davide.diienno-somministrato}@thalesaleniaspace.com,
{daniele.gregori, mattia.paladino}@e4company.com,
{daniele.jahier, alessio.burrello, sara.vinco, gianvito.urgese, maurizio.martina, guido.masera}@polito.it,
{william.fornaciari, federico.reghenzani, andrea.galimberti, davide.zoni, giovanni.agosta}@polimi.it,
{andrea.acquaviva, f.conti}@unibo.it

***Abstract*—Deploying AI-based functionalities to satellites enhances spacecraft autonomy but presents significant challenges for the design of both hardware and software platforms. Advantages of onboard processing include secure and private computing, the reduction of data uplink/downlink requirements, autonomous anomaly detection and recovery, and in general autonomous spacecraft operation. However, edge inference is computationally intensive and has to be performed onboard while sharing resources with other traditional applications, like avionics, Attitude Orbit Control, and signal/telemetry processing. To enable secure and energy-efficient AI-processing capabilities without decreasing the performance of traditional applications, it is necessary to take an holistic approach, jointly optimizing the hardware and software components. To achieve these goals, within the ISOLDE project, a space demonstrator is developed. In this work, the general architecture and main components of the demonstrator are presented, with a particular focus on the AI applications under development. One of the main objectives of the project is the achievement of technological sovereignty, with open hardware and software solutions. Within this project, a new Linux-capable platform based on RISC-V processors and accelerators is developed, enabling the execution of traditional and AI-based workloads on satellites. The project will target hardware and software codesign to optimize the performance of the platform, prioritizing energy efficient, secure and error resilient computing, and will develop a set of applications to demonstrate the capabilities of the platform.**

***Index Terms*—RISC-V, Space Applications, Edge Inference, Satellite Autonomy, Artificial Intelligence**

I. Introduction

The ISOLDE project, funded by the Key Digital Technologies Joint Undertaking (KDT JU), aims to enhance European high-performance RISC-V-based CPUs, achieving functional and non-functional improvements to compete with or surpass proprietary alternatives [1]. Advanced architectures, novel accelerators, and reusable IPs will be developed, forming a robust compute infrastructure for applications in automotive, industrial, and aerospace domains. With the involvement of leading EU companies and semiconductor manufacturers, the project seeks to establish European sovereignty in semiconductors, close the confidence gap, and drive adoption through prototype solutions, documentation, and benchmarks.

Within this project a demonstrator is developed targeting the space use-case, aiming at revolutionizing onboard computational capabilities for space applications by leveraging RISC-V cores and accelerators, integrated with an advanced software layer, to create a hardware and software platform capable of meeting the dual challenges of demanding computing and memory requirements and the harsh environmental constraints of space.

A key innovation of the project is to harness the CVA6 processor family [2] and tensor accelerators to enable processing capabilities closer to the sensor and support the development of onboard inference capabilities. This approach aims to transform traditional workflows, where satellite data, such as Earth observation images, undergo initial preprocessing onboard before being transmitted to ground stations for high-performance computing. By bringing these processing steps onboard, the project not only minimizes uplink and downlink requirements, but also accelerates real-time decision-making and enhances satellite autonomy.

The space demonstrator targets a low-Earth orbit satellite use-case, inspired by the Sentinel-2 mission [3], with an orbiting altitude of ~786 km, and orbital period of ~6036 s. The demonstrator encompasses a broad range of applications, from avionic functionalities and attitude orbit control (AOC) to telemetry processing for fault detection, identification, and recovery (FDIR). Beyond these traditional domains, ISOLDE focuses on artificial intelligence (AI) models to further enhance spacecraft autonomy. These models will be crucial for applications such as hyperspectral data classification and

FDIR. The performance of these AI algorithms will be evaluated on the new hardware, showcasing their potential to redefine the role of onboard computing in space systems.

II. A RISC-V Ecosystem for Onboard AI

A. Overall Hardware Architecture

Figure 1 depicts the heterogeneous architecture adopted for the space demonstrator. The system comprises the Linux-capable Cheshire host platform [4] (delimited by the dashed line), implementing a RV64GC CVA6 core [2]. The RV64GC ISA (RV64IMAFDCZicsr_Zifencei) is selected to support a 64-bit memory space, control status register instructions, compressed instructions, atomic instructions, and hard float/double instructions, the latter necessary to enable complex operations, such as trigonometric or exponential functions, that are not offloaded to specialized accelerators, either because they are rarely occurring during the computation or because the area overhead resulting from an additional dedicated hardware unit would not bring any significant benefit to the performance. An AXI4 crossbar connects [5] the core with the last-level cache, JTAG, direct memory access (DMA), and other peripherals. The open-source platform are selected due to their maturity, support to the Linux OS, and flexibility. The latter is achieved with extensible on-chip (AXI4) and die-to-die interfaces (D2D), important to support multiple domain specific architectures, such as security coprocessors and tensor accelerators, and to enable scalability with multiple chiplets. The demonstrator implements tightly and loosely coupled reconfigurable and mixed-precision accelerators on which are offloaded compute-intensive AI workloads. In particular, two memory mapped accelerators will implement fast and energy efficient vector-vector, matrix-vector, and matrix-matrix operations, which are the backbone of the layers composing the neural network models described in Section III. Both accelerators will implement a reconfigurable spatial array with different architectures and control strategies: one accelerator will be based on a custom architecture, whereas the other uses multiple RISC-V (RV32) cores to process data and control the spatial array.

In the system presented in Figure 1, the CVA6 core executes the OS, general tasks, configures the accelerators, and control the peripherals. The accelerators only execute compute intensive workloads and have their own DMAs and control units, to independently access the system memory and process data.

B. Software Stack

One of the objectives of the space demonstrator is to develop a software stack that enables the automated deployment of optimized AI applications. To achieve this, the hardware-aware software stack of Figure 2 is under development.

Models developed with popular AI frameworks can be compressed and optimized for the accelerator on which are going to be offloaded. Compression is done to reduce the computational complexity and memory footprint, to enable the deployment to resource-constrained edge devices, such as autonomous satellites [6]. Neural architecture search (NAS),

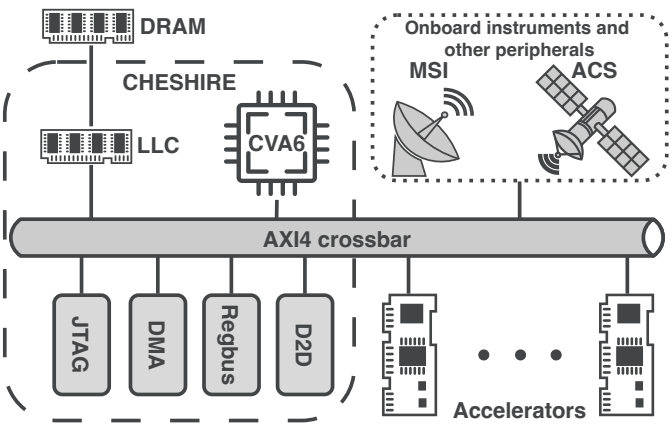

Fig. 1: High-level system architecture of the space demonstrator.

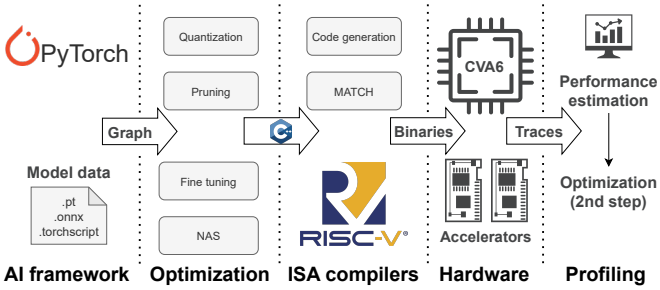

Fig. 2: AI applications software stack.

quantization, and pruning are applied to achieve low-latency and energy-efficient edge inference, leveraging the flexibility of the hardware accelerators supporting multiple precision levels and workload geometries, while fine-tuning is used to retain the task accuracy of the full precision model. Hardware-aware NAS algorithms such as the ones proposed in the open-source framework Plinio [7] are used to fully exploit HW acceleration capabilities, while respecting memory and latency constraints. Once compressed, the AI model is transformed to a representation that can be interpreted and optimized by the compiler, which generates the binaries that are executed or simulated on the space demonstrator. In this step, compilers specifically targeting heterogeneous RISC-V based SoCs such as MATCH can be used to map and optimize the execution of AI workloads to the platform developed for the demonstrator [8]. Extracted performance metrics can be used to further optimize the model, adapting the compression policy.

III. AI Applications Developed for the Space Demonstrator

A. Hotspot detection application

The demonstrator for the space use-case aims to assess the feasibility of performing deep learning inference on a RISC-V processor aboard a satellite. The first application focuses on utilizing AI algorithms to detect wildfires from multispectral satellite images collected from Sentinel-2 satellite. The

THRawS (Thermal Hotspots in Raw Sentinel-2 Data) dataset, available as an open-source resource (THRawS) [9], contains data on various wildfire and volcano events. This dataset includes hyperspectral/multispectral images, where each pixel represents the reflectance of light within specific spectral ranges. Ground truth annotations indicate whether a pixel corresponds to a volcano or wildfire event. Using these labels, a neural network can be trained to identify thermal hotspots effectively.

Before training the neural network, the raw images undergo a preprocessing stage. Specifically, a coregistration process aligns the different spectral bands into a single image for training purposes. To minimize computational overhead, a lightweight and coarse coregistration method (PyRawS) [10] is employed. This approach reduces preprocessing costs significantly.

Additionally, standard preprocessing steps commonly used in ground-based applications are excluded to further reduce computational complexity. Notably, the application relies solely on raw images for inference tasks, bypassing the use of higher-level L1C images typically utilized in on-ground operations.

1) Model Architecture: To train the network, the hyperspectral images are cropped into 7x7 patches and fed into the network. Indeed, this is the size of the patch used to classify the central pixel of each patch in "fire" or "not-fire" classes. This cropping operation is repeated for each pixel of the image to get the segmentation of the entire image.

One of the most used architectures for hyperspectral target detection is the convolutional neural network. Both the 2D and the 3D variants have been successfully employed [11]. Some works even proposed architecture that implements both these variants to fully exploit their features extraction capability [12]. In this work, a 3D convolutional neural network is used. This architecture presents three convolutional blocks. The resulting feature maps are then flattened and passed through a linear layer which predicts the probability of the central pixel of the patch being a target pixel. Figure 3 depicts the scheme of the architecture. In Figure 4 an example of inference result is reported.

2) 3D Convolutional block: Unlike traditional 2D convolutions, which operate on individual spectral bands separately, 3D convolutions can capture spectral and spatial features jointly, providing a more comprehensive representation of the hyperspectral data. By considering the entire spectral cube as a 3D volume, 3D convolutional layers can learn spatial and spectral patterns that are not easily detectable. The proposed convolutional block comprises a 3D convolution layer with a kernel size of $3 \times 3 \times 3$ followed by another 3D convolution layer with a kernel size of $3 \times 1 \times 1$ and stride $2 \times 1 \times 1$ to reduce the spectral dimension. The convolutional block used in this architecture is shown in Figure 3. The number of filters of the first layer is set to a number n while for the second and third convolutional layers, this value increases by $3/4$ of the previous value.

B. Satellite Health Monitoring

The second application is of particular interest in the domain of satellite operations. FDIR is a critical component of satellite systems designed to ensure the reliability and maintain operability of spacecraft throughout its expected lifetime. FDIR processes are established methodologies aimed at identifying malfunctions within a system, isolating the fault, and implementing recovery actions to restore normal operations. However, traditional FDIR approaches require expert knowledge from the specification to the disposal, representing a significant workload throughout the engineering and service phases. The incorporation of AI techniques in the FDIR process represents a transformative approach to satellite health monitoring. The approach presented in this work proposes to improve the fault detection (FD) phase by utilizing machine learning (ML) algorithms to analyze telemetry data onboard the spacecraft. This approach enables the early detection of anomalies and potential faults within satellite systems, thus reducing response times to critical issues.

In the context of this project, the objective is to adapt an implementation of some ML methods to the selected demonstrator architecture, which is based on RISC-V. Deployment on RISC-V processors represents a great opportunity as it may enable the use of hardware acceleration for computationally intensive operations. Traditional Space-Grade processors are not equipped with dedicated accelerators, thus adapting AI-based applications is challenging. Additionally, it must be taken into account that FDIR processes are a critical and fundamental part to maintain spacecraft functionality, thus must reside on the main onboard computer, avoiding the offloading to powerful but less reliable co-processor, and be efficient.

The methods considered for this work include neural network based methods, such as convolutional autoencoders and long short-term memory (LSTM) networks. Autoencoders are deep learning methods featuring a encoder-decoder architecture. In this case, efficient convolutional layers are used. LSTMs are another deep learning method belonging to the family of recurrent neural networks, known to be more demanding from a computational and memory point of view with respect to convolution-based networks. For the proposed methods, it was chosen to follow the principle of semi-supervised learning: all the networks are trained on nominal data (i.e., without anomalies or failures) and then validated against data containing anomalies. In particular, the networks are trained on multivariate time-series from units of a spacecraft subsystem. Considering that data-driven approaches suffers from unbalanced data and requires a large amount of data samples, the anomalies are usually simulated, especially in the initial stage of model exploration and fine-tuning. The data requires a preprocessing step that shall be taken into consideration, as it must be performed onboard the spacecraft. Relevant preprocessing steps include data conversion (e.g., from analog sensors to digital human-readable values), polishing, windowing (i.e., collection of more samples through

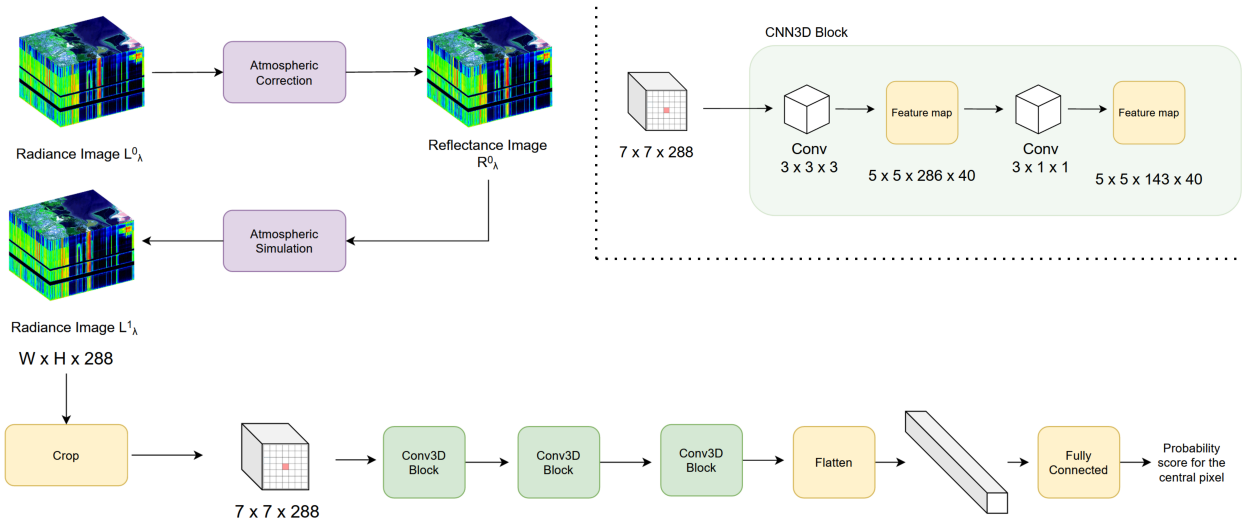

Fig. 3: The architecture of the proposed model. On the top right the 3D convolutional block is depicted.

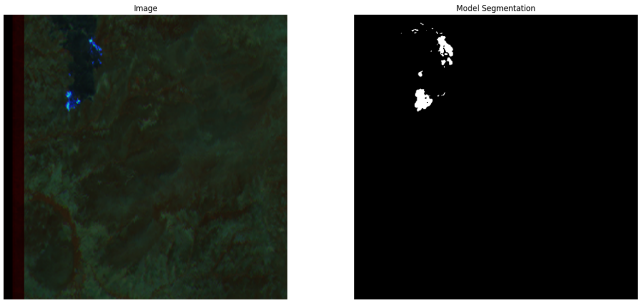

Fig. 4: A test image for the hotspot application. On the left a multispectral image representation. The blue area represents the spreading of a wildfire. On the right, the segmentation result of the model that correctly detect the wildfire area.

time to build a time series of N points) and normalization (e.g., between 0 and 1).

IV. Conclusion

This work presented the ongoing development of the space demonstrator under development for the ISOLDE project, with a particular focus on the AI applications for wildfire detection and satellite health monitoring. The execution of these AI workloads onboard enables autonomous satellite operations, but comes at the cost of high computational complexity, energy consumption, and processing latency. To enable edge inference on a resource constrained environment, a joint hardware and software design approach must be taken. The goal of the space demonstrator is to showcase the potential of this codesign approach applied to an all-European open hardware and software platform.

Acknowledgment

The ISOLDE project, nr. 101112274 is supported by the Chips Joint Undertaking and its members Austria, Czechia, France, Germany, Italy, Romania, Spain, Sweden, Switzerland.

References

[1] "Isolde project." https://www.isolde-project.eu/. Accessed: 2025-01-16.
[2] F. Zaruba and L. Benini, "The cost of application-class processing: Energy and performance analysis of a linux-ready 1.7-ghz 64-bit risc-v core in 22-nm fdsoi technology," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 11, pp. 2629–2640, 2019.
[3] "Sentinel-2 mission." https://sentiwiki.copernicus.eu/web/s2-mission. Accessed: 2025-01-16.
[4] A. Ottaviano, T. Benz, P. Scheffler, and L. Benini, "Cheshire: A lightweight, linux-capable risc-v host platform for domain-specific accelerator plug-in," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 10, pp. 3777–3781, 2023.
[5] "Axi interface." https://docs.openhwgroup.org/projects/cva6-user-manual/01_cva6_user/AXI_Interface.html. Accessed: 2025-01-16.
[6] M. Capra, B. Bussolino, A. Marchisio, G. Masera, M. Martina, and M. Shafique, "Hardware and software optimizations for accelerating deep neural networks: Survey of current trends, challenges, and the road ahead," *IEEE Access*, vol. 8, pp. 225134–225180, 2020.
[7] D. J. Pagliari, M. Risso, B. A. Motetti, and A. Burrello, "Plinio: a user-friendly library of gradient-based methods for complexity-aware dnn optimization," in *2023 Forum on Specification & Design Languages (FDL)*, pp. 1–8, IEEE, 2023.
[8] M. A. Hamdi, F. Daghero, G. M. Sarda, J. Van Delm, A. Symons, L. Benini, M. Verhelst, D. J. Pagliari, and A. Burrello, "Match: Model-aware tvm-based compilation for heterogeneous edge devices," *arXiv preprint arXiv:2410.08855*, 2024.
[9] "Thraws." https://zenodo.org/records/7908728. Accessed: 2025-01-16.
[10] "Pyraws." https://github.com/ESA-PhiLab/PyRawS. Accessed: 2025-01-16.
[11] H. Lee and H. Kwon, "Going deeper with contextual cnn for hyperspectral image classification," *IEEE Transactions on Image Processing*, vol. 26, no. 10, pp. 4843–4855, 2017.
[12] S. K. Roy, G. Krishna, S. R. Dubey, and B. B. Chaudhuri, "Hybridsn: Exploring 3d-2d CNN feature hierarchy for hyperspectral image classification," *CoRR*, vol. abs/1902.06701, 2019.