E2E

Onboard satellite real-time classification of thermal hotspots events on optical raw data

Meoni, Gabriele; Del Prete, Roberto; Ancos-Villa, Lucia; Albalate-Prieto, Enrique; Rijlaarsdam, David; Espinosa-Aranda, Jose Luis; Longépé, Nicolas; Graziano, Maria Daniela; Renga, Alfredo

DOI
10.1007/s42064-024-0249-x

Publication date
2025

Document Version
Final published version

Published in
Astrodynamics

E2E: Onboard satellite real-time classification of thermal hotspots events on optical raw data

Gabriele Meoni[1,2] (✉), Roberto Del Prete[2,3], Lucia Ancos-Villa[4], Enrique Albalate-Prieto[4], David Rijlaarsdam[4], Jose Luis Espinosa-Aranda[4], Nicolas Longépé[2], Maria Daniela Graziano[3], and Alfredo Renga[3]

1. *Department of Space Engineering of the Faculty of Aerospace Engineering, TU Delft, Kluyverweg 1, Delft 2629 HS, the Netherlands*
2. *Φ-Lab, European Space Agency, Via Galileo Galilei 1, Frascati (RM) 00044, Italy*
3. *Department of Industrial Engineering, University of Naples Federico II, P.le Vincenzo Tecchio 80, Napoli 80125, Italy*
4. *Ubotica Technologies, DCU Alpha, Old Finglas Road 11, Glasnevin, Dublin D11KXN4, Ireland*

ABSTRACT

Nowadays, the use of Machine Learning (ML) onboard Earth Observation (EO) satellites has been investigated for a plethora of applications relying on multispectral and hyperspectral imaging. Traditionally, these studies have heavily relied on high-end data products, subjected to extensive pre-processing chains natively designed to be executed on the ground. However, replicating such algorithms onboard EO satellites poses significant challenges due to their computational intensity and need for additional metadata, which are typically unavailable on board. Because of that, current missions exploring onboard ML models implement simplified but still complex processing chains that imitate their on-ground counterparts. Despite these advancements, the potential of ML models to process raw satellite data directly remains largely unexplored. To fill this gap, this paper investigates the feasibility of applying ML models directly to Sentinel-2 raw data to perform thermal hotspot classification. This approach significantly limits the processing steps to simple and lightweight algorithms to achieve real-time processing of data with low power consumption. To this aim, we present an end-to-end (E2E) pipeline to create a binary classification map of Sentinel-2 raw granules, where each point suggests the absence/presence of a thermal anomaly in a square area of 2.5 km. To this aim, lightweight coarse spatial registration is applied to register three different bands, and an EfficientNet-lite0 model is used to perform the classification of the various bands. The trained models achieve an average Matthew's correlation coefficient (MCC) score of 0.854 (on 5 seeds) and a maximum MCC of 0.90 on a geographically tripartite dataset of cropped images from the THRawS dataset. The proposed E2E pipeline is capable of processing a Sentinel-2 granule in 1.8 s and within 6.4 W peak power on a combination of Raspberry PI 4 and CogniSat-XE2 board, demonstrating real-time performance.

KEYWORDS

onboard processing
onboard machine learning
artificial intelligence (AI)
thermal anomalies classification
raw data

Research Article

Received: 21 March 2024
Accepted: 31 October 2024

1 Introduction

In the last years, an increasing number of studies have explored the use of Machine Learning (ML) models for processing mission payload data directly onboard Earth Observation (EO) satellites, initiated by Refs. [1, 2]. Some of the benefits coming from the implementation of onboard Artificial Intelligence (AI) include but are not

✉ gabriele.meoni@esa.int

Nomenclature

AI	Artificial Intelligence	L1C	Level-1C
CNN	Convolutional Neural Network	MCC	Matthew Correlation Coefficient
CPU	Central Processing Unit	ML	Machine Learning
CSC	Coarse Spatial Coregistration	NCS	Neural Compute Stick
CV	Cross-Validation	SHAVE	Streaming Hybrid Architecture Vector Engine
EO	Earth Observation	THRawS	Thermal Hotspots in Raw Sentinel-2 images
FLOAT16	16-bit Floating Point	VPU	Vision Processing Unit
L0	Level-0		

limited to: (a) a reduction of downlink data by discarding the images with limited information content [3, 4], (b) decision-making ability against natural-disasters requiring a near real-time response [5–8], (c) fast retrieval of actionable information, such as changes or targets in local areas [9], (d) implementation of data correction steps such as atmospheric correction [10], and (e) onboard data compression [11, 12].

The Φ-Sat-1 mission [3] marked a pioneering venture as the inaugural spacecraft to incorporate an AI processor on board, specifically the Intel® Movidius™ Myriad™ 2 Vision Processing Unit (VPU). This groundbreaking step led to the genesis of numerous similar missions, with a series of subsequent satellite launches [5, 13, 14]. However, a common prerequisite for all these missions has been the necessity for implementing pre-processing schemes. This requirement is underscored in various studies [5, 8, 9, 15] with the pre-processing methods typically replicating the workflows used on the ground, e.g., geometric and radiometric corrections to the raw satellite data. When applied on board satellites, these solutions often require additional hardware, such as Field Programmable Gate Arrays or Graphics Processing Units, leading to significant payload complexity. It is worth noting that, in the case of Sentinel-2 images, Level-1C (L1C) products or beyond are typically used [5, 8, 9, 11, 12]. However, the quality of L1C images is generally beyond the one achievable on board EO satellites, especially in the case of smallsats, because of the computational complexity of some processing steps and the need for specific information not directly available on board (e.g., Sun intensity, incident angles, path radiance, and others).

In order to facilitate the research on lightweight pre-processing chains, in our previous work [7], we proposed a methodology to design datasets for the detection of events of interest on Sentinel-2 "raw" data, i.e., decompressed Level-0 (L0) with additional metadata. As a result, we designed Thermal Hotspots in Raw Sentinel-2 images (THRawS), a Sentinel-2 raw data dataset for detecting thermal hotspots.

In this work, we capitalised on the previous findings to investigate the suitability of Deep Neural Networks to create a thermal hotspot classification map directly on raw Sentinel-2 granules by limiting the pre-processing only to the proposed Coarse Spatial Coregistration (CSC) technique. The rationale is to enable real-time End-to-End (E2E) processing, i.e., a processing time lower than the image acquisition time, while significantly limiting the payload energy consumption. This is a fundamental aspect of thermal hotspot classification and other natural disaster applications, for which the delays in detection and notification are key parameters to be minimized. Notably, Sentinel-2 raw data were used to demonstrate this concept because additional optical raw data were scarce. However, in principle, the proposed concept can be extended to other raw pushbroom multispectral data imagery.

The novel contributions of the paper in the field of AI-enhanced onboard processing can be summarized as follows:

(1) We develop a classification dataset for thermal hotspots in raw Sentinel-2 imagery. To the authors' knowledge, this is the first dataset of its kind.

(2) By relying on different training strategies, we successfully trained an EfficientNet-lite0 model on Sentinel-2 raw data for thermal hotspot classification, demonstrating the effectiveness of the proposed patch-based detection approach.

(3) We propose an E2E conceptual processing pipeline[①]

① The code implementing the E2E pipeline is available at the following GitHub repository: https://github.com/GabrieleMeoni/END2END.git

that extracts a thermal anomaly classification map from a Sentinel-2 granule through the EfficientNet-lite0 model and lightweight processing.

(4) We implement a breadboard of the E2E pipeline by leveraging different hardware, including the CogniSat-XE2 board, a space-targeted device with flight heritage [16]. We comprehensively measure the pipeline's latency and power consumption, demonstrating real-time performance within 6.4 W peak-power consumption.

The remainder of the paper is organized as follows: Section 2 details the proposed E2E onboard processing chain. Section 3 describes the procedure for training the Convolutional Neural Network (CNN)-based thermal anomaly classifier. Section 4 discusses the breadboard implementation of the proposed E2E chain on multiple edge computing hardware chains and the test-bed to perform timing and power characterization. Section 5 presents our results. Section 6 discusses the results achieved, advantages, limitations, and implications of the proposed method. Finally, Section 7 gives our conclusions.

2 E2E processing pipeline

As suggested by the name "end-to-end", the proposed concept is a payload data processing chain that leverages a CNN model to create a thermal hotspot event binary classification map (event/non-event) of multispectral pushbroom data acquisitions directly with minimal pre-processing steps. The idea behind this approach is minimising the pre-processing steps applied on board a satellite by training CNN models on minimally processed raw data. This way, it is possible to speed up the processing and reduce the energy consumption significantly.

The proposed E2E processing chain is depicted in Fig. 1 and comprises three main components: a CSC module, a raster-tiling engine, and a sequential patch inference module.

In the upcoming subsections, each module will be examined in detail to elucidate the rationale behind its incorporation into the processing pipeline.

In this study, the E2E concept is demonstrated on Sentinel-2 raw data. Specifically, the E2E pipeline takes as input Sentinel-2 raw "granules", i.e., the data portion of 23 km × 25 km obtained by a single sensing detector in 3.6 s [17]. From each granule, we select bands B_{8A}, B_{11}, and B_{12} as the standard utilised by numerous previous works to detect thermal anomalies [7, 18–23].

2.1 Coarse coregistration module

Given the pushbroom nature of Sentinel-2 or other pushbroom detectors, spectral bands are not natively overlapped, namely spatially registered. Non-coregistered bands can lead to inaccuracies in techniques that rely on pixel-wise information from multiple spectral bands. Therefore, in the context of thermal hotspot detection, it is essential to coregister these bands accurately for precise and reliable interpretation of the data.

Generally speaking, the process of coregistration can be divided into two distinct stages: a coarse coregistration, aiming for an accuracy of approximately 1 pixel, and a fine coregistration applied to determine any residual transformation that might be needed. The first stage sets the foundational alignment of the bands. In contrast, the second one involves formulating and applying transformation equations, representing the most intensive computational part of the registration

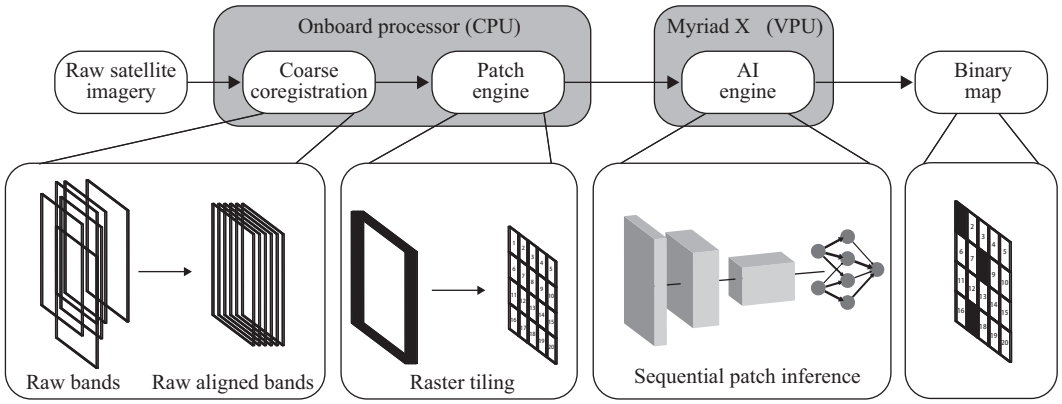


Fig. 1 Developed E2E pipeline for onboard thermal hotspot heatmap classification of Sentinel-2's raw data.

process. In our study, we concentrate only on the first kind of coregistration, prioritizing achieving a general approximate level of alignment. This approach is based on the generalizability of Deep Learning in addressing small misalignment errors [24]. For these reasons, we adopt the CSC technique proposed in our previous work [7] that consists of applying a set of precalculated along-track and across-track shifts $\overline{S_{B_{11}\text{-}B_{8A}}}$ and $\overline{S_{B_{12}\text{-}B_{8A}}}$ to respectively align the B_{11} and B_{12} bands to the reference band B_{8A}. Such shifts $\overline{S_{B_x\text{-}B_y}}$ were calculated by measuring the average along-track and across-track displacements between the bands B_x and B_y of raw granules of the THRawS dataset. In each granule, such displacements were measured as the average distance in pixels of the keypoints extracted and matched through the SuperGlue neural network [25].

2.2 Raster tiling engine

After selecting only the B_{8A}, B_{11}, B_{12} bands, a Sentinel-2 granule, which is 1152 × 1296 wide, is tiled into 256 px × 256 px patches. This is needed to match onboard hardware's memory requirements and distinguish between local features [3, 26, 27]. Notably, the raster tiling operation is performed with no overlap between consecutive patches.

2.3 Sequential patch inference module

The sequential patch inference module operates by taking the tiles created by the patch engine module as input. Each tile is singularly classified by the CNN into thermal-hotspot event/non-event. The outputs for different patches are sequentially collected in a binary classification map, a matrix whose elements contain the assigned labels for each analysed patch. This targeted approach is particularly advantageous for maintaining significant dimensions of each tile. It is worth noting that the ability to preserve tile dimensions ensures that these small yet critical events are not overlooked or inadequately processed, thereby improving the overall accuracy and robustness of the model. This enhancement is especially convenient when dealing with small-scale fires.

The adopted CNN model is an EfficientNet-lite0, which represents an optimized version of the original EfficientNet model [28] for implementations on embedded systems. The choice of the CNN model was driven by the need to reach high classification performance while keeping the number of floating point operations and model size limited because of the need to reach a real-time implementation. The efficacy of EfficientNet models for classification purposes on Sentinel-2 images was already demonstrated in our previous work MSMatch [29] work. The latter also shows a comparison in performances of the different EfficientNet models (i.e., EfficientNet-B0–EfficientNet-B7), demonstrating how the use of more computationally expensive models compared to EfficientNet-B0 leads to a slight increase in classification accuracy at the expenses of a significantly higher number of operations. Because of that, we selected the correspondent lite version of EfficientNet-B0, i.e., EfficientNet-lite0, to minimize the number of floating-point operations. Notably, the effectiveness of EfficientNet-lite0 models for onboard satellite classification was already demonstrated in the scope of "The OPS-SAT case" challenge in-orbit test campaign [27].

3 Dataset creation and model training

3.1 Dataset creation

In order to train our classification model, we needed a dataset specifically designed for thermal hotspot classification. Mainly for this reason, we modified the data from THRawS [7] to suit the requirements of patch classification. THRawS is made of Sentinel-2 raw granules containing volcanic eruptions and wildfires. In each raw granule, thermal hotspots are marked through a bounding box that allows for their localisation inside the granule.

To create our patch classification dataset, we first selected the bands B_{8A}, B_{11}, and B_{12} of each raw granule as proposed for the THRawS dataset. Subsequently, to train a model with the same input as for the E2E pipeline, we applied the CSC technique and tiled the registered bands into 256 px × 256 px patches. This tiling was performed with a deliberate overlap of 10%, ensuring that contiguous areas are not missed and allowing for a more comprehensive analysis of the granules. Upon completion, the dataset was substantial, comprising over 5000 samples, resulting in a wide-ranging and extensive collection of data.

Concerning data labelling, we implemented a categorization system for the analyzed patches based on thermal hotspots, whose bounding boxes are provided in THRawS, on the cropped patches. Specifically, we

designated any patch encompassing a minimum of 9 pixels exhibiting these hotspots as an "event" class. The threshold was set empirically to identify areas with significant thermal irregularities. Conversely, patches that did not meet this criterion, lacking the specified extent of thermal hotspots, were systematically marked as "non-events". This distinction allowed us to differentiate between areas of potential interest and those with normal thermal patterns, facilitating a more focused analysis of thermal irregularities in our data. After tiling, patches were visually inspected to avoid errors in the labeling procedure.

As compiled in this manner, the dataset demonstrates a significant imbalance in its composition, reflecting a notable disproportion in the representation of its classes. Out of the total number of patches analyzed, 4636 were categorized as "non-events", overshadowing the mere 394 patches identified as "events". This latter group constitutes only 7.83% of the entire dataset. Such a disparity in distribution between event and non-event patches highlights a significant imbalance, indicating a predominant prevalence of non-event occurrences within the dataset. This skewness in data could have implications for further analysis, affecting our choices in terms of the training strategy adopted and metrics used to evaluate the classification performance of our CNN model, as described in Section 3.2. After creating our patch classification dataset, we statically split it into train and test by using 90% and 10% splits. We used stratified sampling to keep the exact percentages of the event and non-event classes in the train and test datasets. However, during the stratification sampling process, we split the dataset using a geographical split criterion instead of performing random sampling. This means that all the patches related to a specific dataset location were placed exclusively in the train or the test partition. The idea of such a choice is to avoid data leakages between train and test due to possible spatial similarities of patches with the exact locations that could lead to overestimating the model performances [27]. After tiling, patches were visually inspected to avoid errors in the labeling procedure. The total number of patches in the train and test partitions is 4502 and 531, respectively.

During the model training, the train partition is split at runtime into Train and Cross-Validation (CV) with 90% and 10% split ratio by using a random geographically split logic, performed as shown in Fig. 2. In particular, we first selected a desired Cross-split ratio (δ_{sr}) (e.g., 10%), which fixes the maximum number of CV patches and a value to be used as an initial seed for the next random assignment operations. Then, we create a dictionary containing the name of a location (key) and the corresponding number of patches (value) for partitioning all the volcano, fire, and non-event patches in the training dataset. After randomly shuffling the location names in the dictionaries, we pick the following location for volcano events and assign all the corresponding patches to the CV split. If the number of patches assigned to the CV partition is lower than the maximum, we perform the same step for fire events. In this way, both the CV and the Train partitions will have wildfire and volcanic eruption events. We continue iteratively until the number of CV patches is greater or equal to the maximum; the remainder is assigned to the Train partition. The total number in the CV split is generally overestimated by proceeding this way. Therefore, we calculate the actual split ratio (α_{sr}) and compare it to the desired one. If $|\alpha_{\mathrm{sr}} - \delta_{\mathrm{sr}}| < 0.01 \cdot \delta_{\mathrm{sr}}$, we accept the performed split. On the contrary, if that

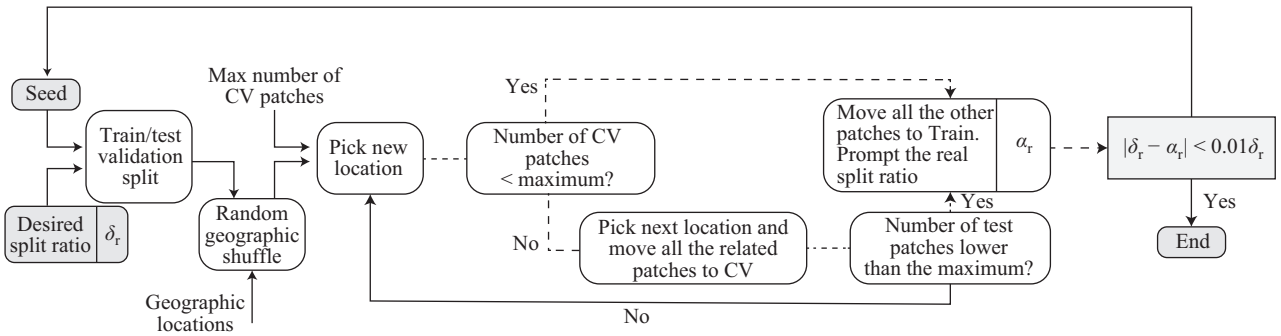


Fig. 2 Pictorial illustration of the method used for dividing the dataset into training and cross-validation partitions. It involves an iterative process that includes several key steps: (1) determining the ratio for the split, (2) performing a random geographical division of the data, and (3) comparing the number of patches in each partition.

condition is not met, we choose another seed so that the random shuffling of the locations will lead to different results, and the split will be performed again.

Using a runtime random assignment of the locations, it is possible to test the effect of the different splits in the Train/CV partitions on the final model performance while keeping the advantages of geographical splitting.

Table 1 shows the number of patches for the different dataset sets and classes.

Table 1 Thermal hotspots dataset overview

Description	Count	% of total dataset
Event patches	394	7.83%
Non-event patches	4636	92.17%
Total patches	5033	100%
Train-validation	Event (360), Non-event (4142)	89.45%
Test	Event (34), Non-event (497)	10%

3.2 Model training

Since both the train and the test datasets are unbalanced, using accuracy as a classification metric is not advisable [30]. Therefore, we opted for Matthew Correlation Coefficient (MCC) as the metric to evaluate and compare our trained models, which already found numerous applications for assessing binary classification problems on unbalanced datasets [30–32]. The MCC can be calculated as shown in Eq. (1) [32]:

$$\mathrm{MCC} \triangleq \frac{T_\mathrm{N} \cdot T_\mathrm{P} - F_\mathrm{P} \cdot F_\mathrm{N}}{\sqrt{(T_\mathrm{P}+F_\mathrm{P})(T_\mathrm{P}+F_\mathrm{N})(T_\mathrm{N}+F_\mathrm{N})(T_\mathrm{N}+F_\mathrm{P})}} \tag{1}$$

where T_P, T_N, F_P, and F_N are, respectively, the total number of True Positives, the total number of True Negatives, the total number of False Positives, and the total number of False Negatives. Since in our dataset, the total number of events is much smaller than the number of non-events, one model predicting all the patches as non-event would have an accuracy roughly of 93.6% on the test set but an MCC of 0, which represents a random prediction. Because of that, MCC represents a valid alternative to scoring the capability of a trained model to predict both event and non-event patches correctly.

However, to partially fix the problem of class unbalance, during training, we upsampled the "event" class of a factor N_UPS by presenting the same patch multiple times after applying a random light augmentation. This allowed us to balance the dataset classes partially. In particular, we tested different values of $N_\mathrm{UPS} \in \{2, 3, 4, 6, 7\}$. We repeated the training for five seeds (i.e., 0, 9, 14, 18, 19) for each value to make a robust performance assessment.

After training, we picked the model featuring the best MCC value on the cross-validation dataset for each of these sets of hyperparameters.

We trained our model using different training strategies, i.e., supervised and MSMatch. We used an EfficientNet-lite0 model pre-trained on ImageNet [33] for both training strategies. To train the models with supervised learning, we applied the weak augmentations included in the MSMatch pipeline [29], i.e., image translations by up to 12.5% and horizontal flips.

The training was performed using stochastic gradient descent with a momentum of 0.9 and weight decay of 0.00075, with an initial learning rate of 0.03. Furthermore, we used a batch size of 8 patches. During training, the CV dataset was evaluated with an exponential moving average with a momentum of 0.999. We noticed that the training process failed for specific combinations of N_UPS and seeds, leading the model under training to predict *non-event* for all the possible inputs. We supposed this problem to be linked to the significant imbalances in the training class. Therefore, we added the weighting factors to the loss function for each class in Eq. (2):

$$\begin{cases} W_\mathrm{E} = \dfrac{4636}{N_\mathrm{UPS} \times 5033} \\ W_\mathrm{NE} = \dfrac{397 \times N_\mathrm{UPS}}{5033} \end{cases} \tag{2}$$

where w_E and w_NE are, respectively, the weighting factors for the event and non-event class. As one can see, when $N_\mathrm{UPS} = 1$, w_E and w_NE weight errors in the loss function proportionally to the inverse of their population percentages in the dataset. When $N_\mathrm{UPS} \neq 1$, w_E and w_NE are scaled to compensate for the effect of the event class upsampling.

Compared to the original work [29], as for supervised learning, we modified the pipeline by adding the "event" class upsampling. After the upsampling, we used 400 samples per class as labelled examples, while the rest were used as unlabelled ones. In addition, compared to the MSMatch pipeline, we did not apply channel normalization. As stated, we used the same setup for the hyper-parameters shared with supervised learning. Furthermore, for each of the eight labelled examples in the batch, we used four unlabelled examples (i.e.,

unlabeled-ratio of 4) and a pseudo-label cutoff threshold of 0.95.

4 E2E breadboard and benchmarking on multiple edge computing hardware

We created a breadboard implementation on target edge computing hardware to measure the timing and energy performance that E2E can attain. In particular, we compared three test setups leveraging different hardware chains:

- E2E pipeline on Raspberry Pi 4 only. The full E2E pipeline is performed on a Raspberry PI device, which was chosen due to its broad use in CubeSat missions [34–36]. Specifically, we used a Raspberry PI 4 equipped with 4 GB of random access memory.
- E2E pipeline on Raspberry Pi 4 and a Neural Comput Stick (NCS) 2[②]. In this setup, the CSC and raster-tiling operations are implemented on the Raspberry Pi 4 Central Processing Unit (CPU), while the inference of the CNN is offloaded to a NCS 2, which is connected to the Raspberry PI CPU through a USB cable. Differently from a Raspberry PI4, which is a general-purpose CPU, the NCS 2 is a processing device specifically tailored for ML applications. Specifically, it is equipped with an Intel® Movidius™ Myriad™ X VPU, a 16 nm AI processor featuring 16 Very Long Instruction Word Streaming Hybrid Architecture Vector Engine (SHAVE) processors, 2 Neural Compute Engines specific for neural network executions and high throughput intelligent memory fabric to reach up to 4 Trillion Operations per second with a maximum power consumption of 1.5 W [37][③]. Because of that, this setup is supposed to outperform the first scenario in terms of timing and energy-performance at the expenses of an increased hardware complexity. The NCS 2 enables quick iterations and is readily available to consumers but is not space grade. This benchmark shows the relevance of testing on NCS 2 for space applications and characterizes the delta between actual space hardware and this Commercial Off-The-Shelf NCS 2.
- E2E pipeline on Raspberry Pi 4 and a Ubotica CogniSat-XE2 board. In this experimental configuration, the CNN model is inferred on the CogniSat-XE2[④]. Similar to the NCS 2, the CogniSAT-XE2 board incorporates a Myriad X VPU. However, differently from the NCS 2, it is specifically designed to accelerate the inference of ML models on CubeSat payloads. It has been successfully deployed in the context of the CogniSat-6 mission [16]. Consequently, this setup provides a lower reality gap solution than the setup using the NCS2.

For all the configurations, we utilised the CNN model with seed = 18 and $N_{\mathrm{UPS}} = 3$, which showcases the best MCC performance on the test set MCC = 0.902. Specifically, to comply with the Myriad X-supported arithmetics, we quantized the trained model via post-training quantization using 16-bit Floating Point (FLOAT16). Indeed, although the INT8 arithmetic could have further improved energy and timing performance, we opted for a floating point format to minimize the implementation loss due to quantization. This choice is also supported by the fact that FLOAT16 is sufficient to achieve real-time performance with no need for further optimization in terms of energy and timing performance. To make a fair comparison, the FLOAT16 quantized model was also used for the setup relying only on Raspberry PI.

For the setups using the NCS2 and CogniSat-XE2 board, the quantized model was then compiled by using *OpenVINO v2022.1*[⑤] before deployment.

The total workflow adopted workflow needed to implement the model on Myriad X is depicted in Fig. 3.

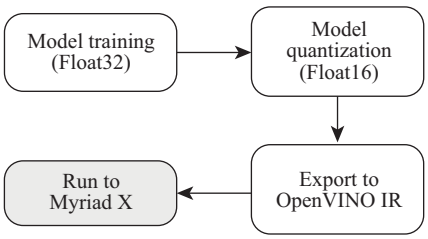


Fig. 3 Pictorial representation of the workflow used to implement a model on Myriad X.

The Siglent SPD3303X Programmable DC Power Supply was used to measure the power consumption during the inference. Timing measurements were

② Intel® Neural Compute Stick 2 (Intel® NCS2): https://www.intel.com/content/www/us/en/developer/articles/tool/neural-compute-stick.html

③ Intel® Movidius™ Myriad™ X Vision Processing Unit: https://www.intel.com/content/www/us/en/ark/products/series/213842/intel-movidius-myriad-x-vision-processing-units.html

④ Cognisat-XE2: https://ubotica.com/ubotica-cognisat-xe2/

⑤ OpenVINO™ toolkit: https://www.intel.com/content/www/us/en/developer/tools/openvino-toolkit/download.html?VERSION=v_2023_2_0&OP_SYSTEM=WINDOWS&DISTRIBUTION=ARCHIVE

performed using *time* Python libraries and recorded on the Raspberry Pi. The full test setup leveraging the CogniSat-XE2 board is shown in Fig. 4.

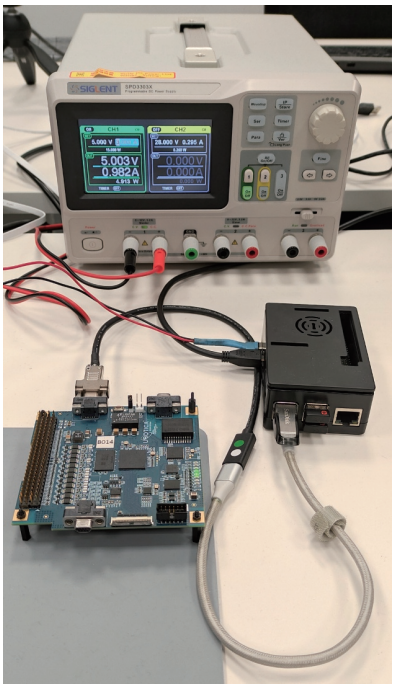


Fig. 4 Test setup. Shown are the Siglent SPD3303X Programmable DC Power Supply, the Raspberry Pi 4, and the CogniSat-XE2 board. The monitor, keyboard, and mouse attached to the Raspberry Pi are not shown.

5 Results

5.1 Training results

The current overview of Table 2 aims to illustrate the model's performance across various training strategies, thereby providing a comprehensive demonstration of its general capabilities.

Indeed, under the previously outlined methodology, we conduct a rigorous benchmarking analysis on the model trained to utilize three distinct approaches, namely MSMatch, Supervised Unweighted, and Supervised Weighted. Based on the observations depicted in Table 2, each graph illustrates the outcomes obtained from five distinct seeds (0, 9, 14, 18, 19), signifying various model initializations across a range of five distinct upsampling factors (2, 3, 4, 6, 7). Such results were obtained by testing the trained models on the test dataset.

Concerning the MSMatch training (Table 2), the range of MCC values spans from −0.2121 to 0.8864. Predominantly, the results showcase positive values, suggesting a reasonable prediction accuracy across most of the setups. However, a significant outlier is observed with the negative MCC value of −0.2121 for the combination of seed 14 and an upsampling factor of 2. This particular result is concerning as it indicates a lack of predictive accuracy and a possible inversion in prediction, where the model may be consistently incorrect. This could point to a potential issue in the model's training process or a specific anomaly in the data for this configuration.

The MCC values in the unweighted supervised training (Table 2) vary from 0 to 0.9023. There are predominantly positive values, indicating effective model performance in most scenarios. However, for both seed 18 and seed 9 with $N_{\text{UPS}} = 2$, the trained model predicts *non-event* for any input patch, which leads to an MCC of 0. Nevertheless, as shown in Table 2, the proposed weighting strategy described in Section 3 is effective in removing such for seeds 9 and 18, leading to positive MCC values. A null MCC is found for seed 19 for $N_{\text{UPS}} = 2$ when using the weighted training strategy. It is interesting that negative or null MCC values are found only for $N_{\text{UPS}} = 2$ for all the different training strategies. That remarks the need for upsampling the underrepresented *event* class during training.

Since our goal is to train the model for any seed

Table 2 Comprehensive evaluation of MCC across different training techniques: MSMatch training, weighted supervised training, and unweighted supervised training, calculated for different upsampling factor values N_{UPS} (2, 3, 4, 6, 7) and seeds (0, 9, 14, 18, 19)

	MSMatch training					Weighted supervised training					Unweighted supervised training				
	2	3	4	6	7	2	3	4	6	7	2	3	4	6	7
0	0.836	0.788	0.839	0.812	0.750	0.795	0.750	0.805	0.793	0.793	0.830	0.788	0.756	0.825	0.812
9	0.886	0.851	0.858	0.526	0.868	0.722	0.869	0.886	0.807	0.871	0.000	0.833	0.774	0.821	0.795
14	−0.212	0.705	0.504	0.812	0.843	0.716	0.817	0.716	0.857	0.902	0.786	0.832	0.871	0.871	0.832
18	0.869	0.839	0.664	0.871	0.805	0.812	0.886	0.830	0.857	0.819	0.000	0.902	0.807	0.854	0.854
19	0.758	0.836	0.793	0.812	0.825	0.000	0.774	0.756	0.830	0.815	0.750	0.851	0.805	0.768	0.833

successfully and not to compare different training techniques to characterize our models' performance, we picked the best model obtained for each combination of (N_{UPS}, seed) listed in Table 2 among the three training strategies, and we averaged them among the seed values. The obtained results, displayed in Table 3, range from 0.780 to 0.854 for N_{UPS} and show an increasing trend for growing values of N_{UPS}.

Figure 5 showcases the classification masks produced by the E2E pipeline on three different raw granules when the best model is used (MCC = 0.902). The classification mask produced for the first granule, correspondent to a wildfire acquired in Australia in 2019—coordinates (lat = −29.92°, lon = 152.34°)—underlines the capability of the trained to detect events much smaller than the patch size and partially affected by registration errors due to the coarse spatial registration technique used. This fact seems to remark on the findings by Fanizze *et al.* [24], which states that the robustness of the model to registration errors can be increased by including misregistered patches during training.

Table 3 Average MCC results over different training seeds for different upsampling factor values (2, 3, 4, 6, 7). Such results were achieved by averaging over seeds the best value for each (N_{UPS}, seed) among the three training strategies

N_{UPS} (average over seed)				
2	3	4	6	7
0.780	0.848	0.839	0.8408	0.854

To further investigate the impact of coregistration

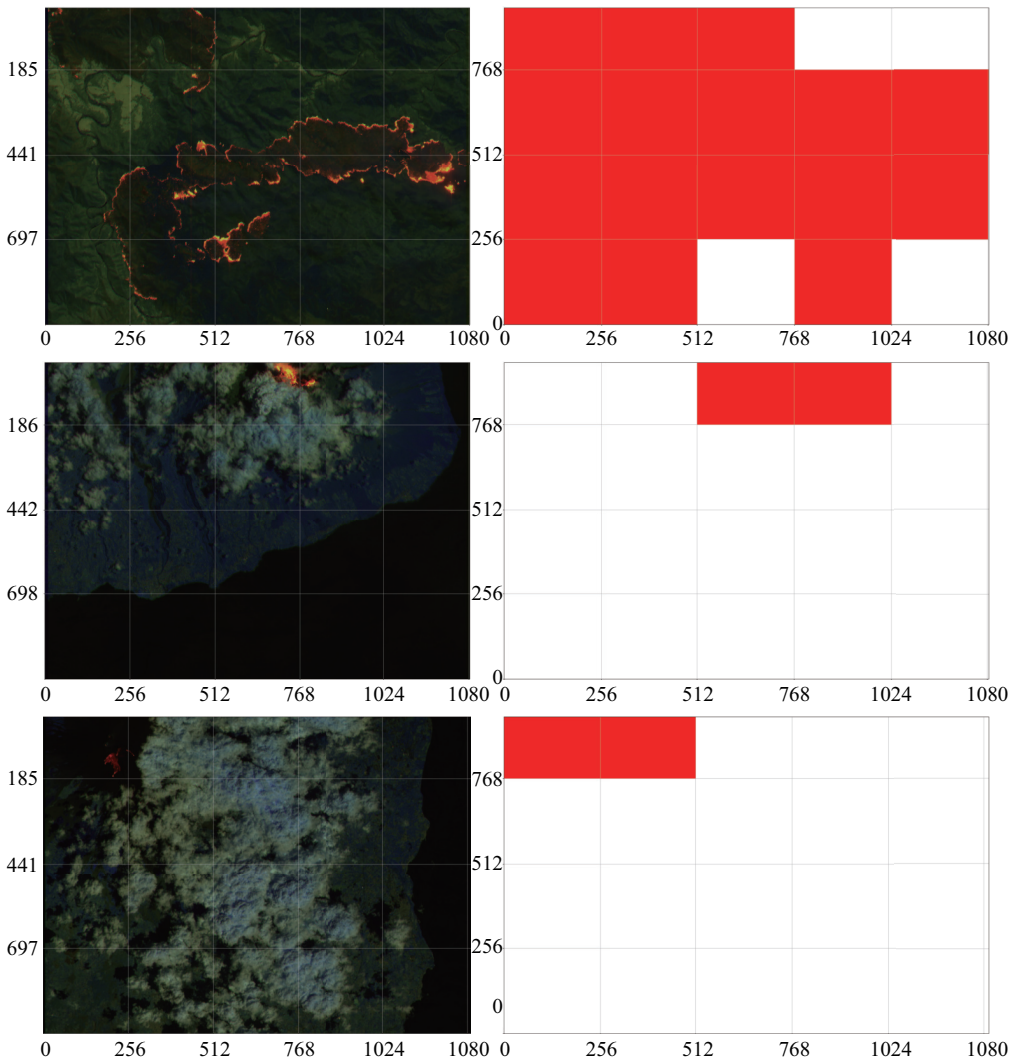


Fig. 5 Classification masks for three different raw granules produced by the E2E pipeline when the best model (MCC = 0.902) is used. Patches classified as *event* are marked in red.

errors on possible misclassifications, we checked the misclassified patches—Fig. 6—by the best model on the test split. Two of the six misclassified patches feature events of small size that are completely or significantly unregistered (i.e., patches 2, and 3). In these cases, it is possible to suppose that the entity of the registration errors, significantly higher than the average value of 1 px reported in our previous work [7], combined with the presence of events of very reduced size could have led the model to classify those patches as *non-events* wrongly.

In patches 1 and 5, registration errors are not evident. However, both these patches feature a significant presence of clouds/smoke partially overlapping with the thermal hotspot event, which could have contributed to the classification error. Patch 4 is difficult to judge by visual inspection, since it is difficult to distinguish events from other unregistered elements in the image.

Finally, patch 6 is predicted by the model as *event* despite its expected label being *non-event*. However, it is possible to notice by visual inspection that the model's prediction is correct, which showcases that the model outperformed our labelling procedure for this patch.

5.2 Power consumption and latency results

All the setup were run for 3 granules for 256 iterations. Moreover, in the case of the Raspberry Pi + CogniSat-XE2 setup, the experiments were performed by running one and two executors on the CogniSat-XE2 device. An executor consists of a mechanism of the board firmware that interacts with the SHAVE processors to run the inference phase in parallel by adding two instances of the same neural network, making it possible to parallelize two instances of the same neural network or two different models.

This normally improves the execution time when several inferences can be run in parallel but also adds a small overhead related to the synchronization of the operations performed on the board. A summary of the results can be seen in Table 4, showing the average and peak power (W) of each execution and the mean running time per granule considering the whole process. It is worth noting that using both the NCS2 or the CogniSat-XE2 board allows the processing of the granules with real-time performance, i.e., with a processing time lower than the acquisition time (3.6 s) [17]. On the contrary, the setup leveraging only a Raspberry Pi 4 surpasses the required time per granule by far. A more detailed explanation of each of the results is featured below.

5.2.1 Raspberry Pi 4

The execution of the Raspberry Pi 4 was carried out to establish a baseline and investigate the possibility of running the proposed E2E chain in real time on satellite hardware.

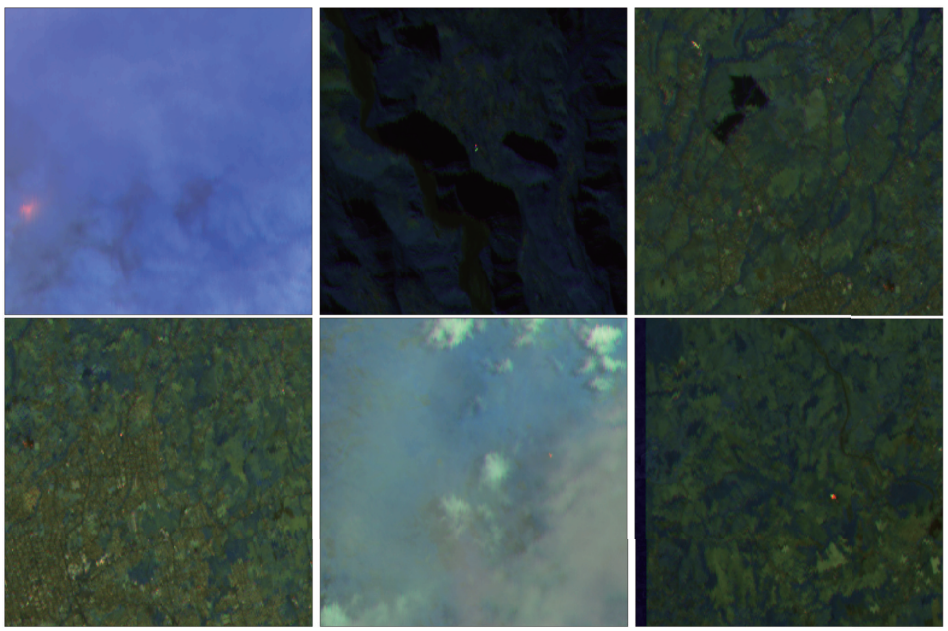


Fig. 6 Misclassified patches by best model (MCC = 0.902) on the test dataset. Starting from the top left, we number the patches 1 to 6.

Table 4 Results – power and timing benchmark measurements

Deployment device	Average power (W)	Peak power (W)	Time per granule (s)
Raspberry Pi 4	5.3	6.2	16.2
Raspberry Pi 4 & NCS2	5.5	6.6	1.6
Raspberry Pi + CogniSat-XE2 (1 executor)	5.8	6.4	2.0
Raspberry Pi + CogniSat-XE2 (2 executors)	5.5	6.4	1.8

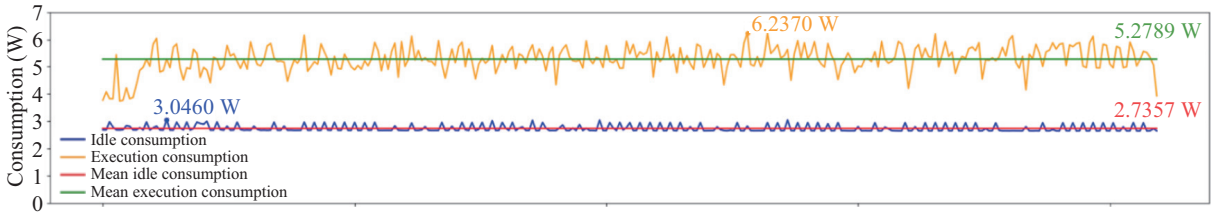


Fig. 7 Graph displaying the Raspberry Pi 4 power consumption over time, with blue indicating idle and yellow for execution consumption. Horizontal lines represent average values, showing 5.27 W during execution and 2.73 W in idle state.

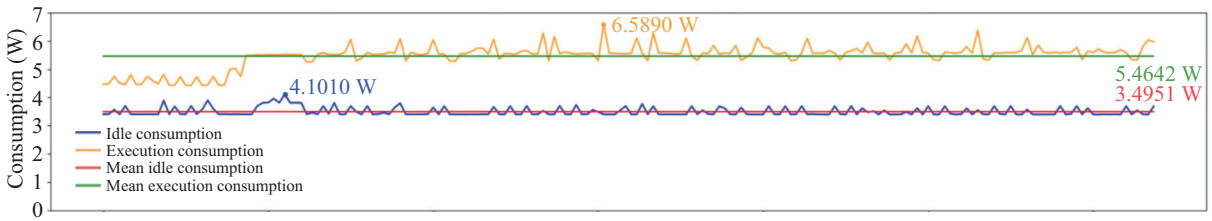


Fig. 8 Graph displaying the Raspberry Pi 4 + NCS2's power consumption over time, with blue indicating idle and yellow for execution consumption. Horizontal lines represent average values, showing 5.46 W during execution and 3.49 W in idle state.

The whole execution took 16.2 s per granule on average, while the execution had a peak power consumption of 6.2 W and a mean power consumption of 5.3 W. Notably, this configuration does not enable the execution of our pipeline in real time since the processing time is significantly higher than the acquisition time.

Figure 7 depicts the power consumption evolution during the whole process (in yellow) and compares it with the consumption of the system in idle mode in similar circumstances.

5.2.2 Raspberry Pi 4 & NCS2

Adding the NCS2 to the setup allows reducing the computational time to 1.6 s per granule, allowing real-time execution. Moreover, the execution had a peak power consumption of 6.6 W and a mean power consumption of 5.5 W.

Figure 8 depicts the power consumption evolution during the whole process (in yellow) and compares it with the consumption of the system in idle mode in similar circumstances. It is worth noting that the initial low consumption of the system is due to the initial steps to set up and run the NCS2 device without performing inference.

5.2.3 Raspberry Pi + CogniSat-XE2

To test with the CogniSat-XE2 board, two different configurations were run, using 1 and 2 executors. For both cases, the pre-processing required prior to running the inference was run on the Raspberry Pi, with a computational time of 0.7 s.

Configured to run with one executor, the inference time for 3 granules is 5.3 s, making a total of 6 s, inclusive of the pre-processing time, obtaining an average inference time of 2.0 s per granule, which will allow real-time execution. With this configuration, the average and peak power consumption reported for the whole system are respectively 5.8 and 6.4 W (Fig. 9).

In addition, with two executors, the inference time for 3 granules is 4.7 s. When the pre-processing time is considered, the total processing time is 5.4 s, which corresponds to an average total processing time of 1.8 s. When two executors are used, the average and peak power consumption were respectively 5.5 and 6.4 W (Fig. 9).

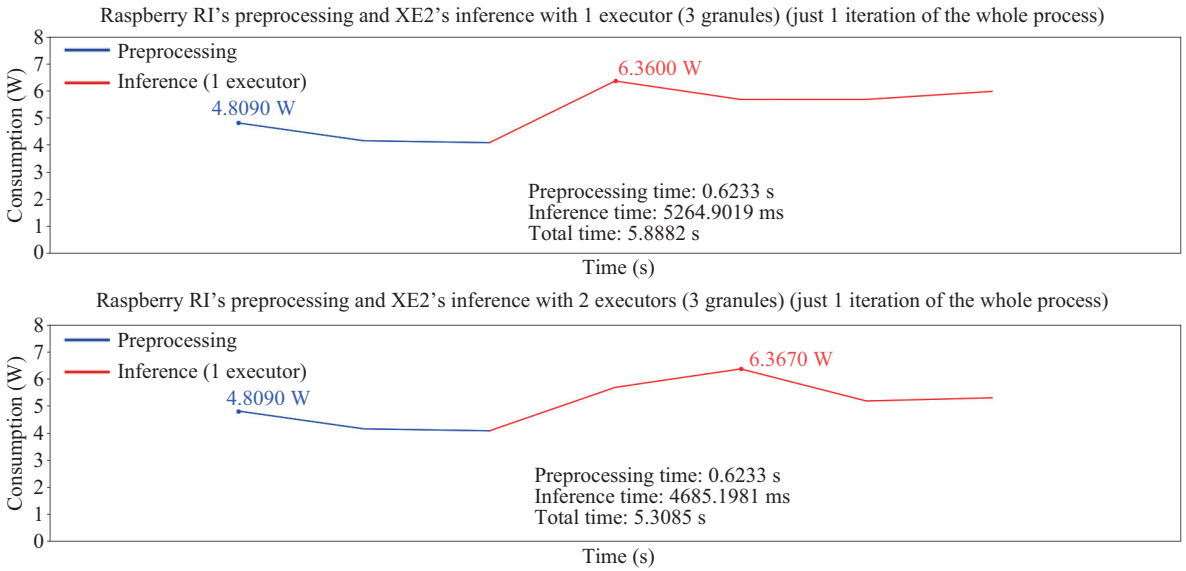


Fig. 9 Graphs illustrating power consumption over time for the Raspberry Pi 4 + CogniSat-XE2 are presented: the top plot shows usage with a single executor and the bottom with dual executors. Blue marks the pre-processing phase, while red denotes the inference stage. This assessment was conducted using three granules, limited to a singular iteration each.

6 Discussion

Previous works have advocated for the use of AI to pre-filter data directly on the spacecraft to move away from the "first-in-first-out" bent-pipe model [37, 38]. However, existing work, such as the work by Furano *et al.* [37] and by Giuffrida *et al.* [3], proposes the utilisation of a file storage system in which files are stored, optionally pre-processed, and subsequently pulled for AI interpretation. While this approach can provide value, the lightweight methodology proposed here is distinct from these storage and pre-processing-based approaches. By operating on raw data, the pipeline proposed here is lightweight while maintaining sufficient accuracy. Since the system is able to process data at a faster rate than the ones data are produced, the need to store the information in storage is mitigated, and data can be transferred directly to the AI accelerator for inference. Notably, performing processing in real time allows for significantly curtail the delay in the creation of the alert map compared to existing on-board AI processing approaches, which are slower due to the need for storage and/or pre-processing of images before AI inference can commence. This is critical for thermal hotspot classification or other early alert systems for which the delay in data processing directly affects the delay in the notification of target alerts to end-users. One limitation of our analysis is the assumption that the image to be processed is immediately available after its creation. Consequently, we are overlooking potential delays associated with transferring images from the camera to the payload mass memory and from the mass memory to RAM. However, some implementations of very high-speed communication protocols, such as SpaceFibre [39], can attain a useful data-rate of 2.5 Gb/s [40]. Because of that, considering that the size of an entire raw image is roughly 0.38 Gb, the delay due to such data transfers can be reduced to less than 0.2 s. Because of that, this assumption does not substantially impact our results.

Another important result is the significantly reduced energy and peak power consumption during the model inference. Indeed, the energy consumption directly affects the available orbit power and, therefore, the maximum duty cycle and coverage area that can be attained. Moreover, the E2E pipeline operates assuming that the payload imager and the payload data handling unit operate simultaneously. Because of that, reducing the peak power consumption is of utmost importance to ensure the heat generated can be adequately dissipated. Therefore, having a low peak and energy consumption is critical to ensure the scalability and the miniaturization of payload design and, therefore, limit the mission cost.

Our results show that the real-time speed of data processing and energy efficiency are achievable thanks to two main factors. Primarily, the inclusion of an

AI hardware accelerator in the system architecture is of utmost importance. Without this accelerator, the required throughput is not achieved in our setup. Moreover, a significantly larger amount of energy is consumed. Therefore, a design for any space system that incorporates this paradigm should be optimised for this trade-off of system complexity, throughput, energy consumption, hardware mass and volume. It is likely that for smallsats with similar ARM-based processing, the inclusion of a hardware accelerator in the system architecture is required. The other factor that enables real-time processing is the adoption of a lightweight pre-processing, consisting of the CSC proposed in our previous work [7]. The latter ensures sufficient accuracy even in cases in which events significantly smaller than the patch size are included. Notably, some exceptions exist in which the entity of registration errors is significantly higher than the average, leading to a complete or significant displacement of small thermal hotspots. Despite these cases representing exceptions, their correct detection is of utmost importance for early alert systems. Because of that, future works will focus on improving the timing/quality trade-off of the bandwidth registration, exploring the investigation of ML-based models. Finally, it is worth remarking that the used "raw" data, as defined in our previous work [7], differs from the Sentinel-2 sensory data because of the loss of information introduced by the compression scheme adopted in the Sentinel-2 mission [41], and not-reverted onboard equalization.

The latter was not reverted because of the impossibility of retrieving the coefficients applied to each granule. If onboard equalization is necessary to ensure sufficient model performance, it could be included in the proposed pipeline scheme without significantly affecting our real-time performance, being it a lightweight processing step (i.e., double-step piece-wise linear function). Therefore, we believe these discrepancies between the used data and real sensory-raw data do not affect the validity of our results.

7 Conclusions

The study has proposed a payload data processing concept to create a thermal hotspot binary classification map from processing multispectral or hyperspectral raw data end to end. To this aim, data are provided as input to a CNN after being pre-processed through a tailored lightweight pre-processing strategy. Extensive benchmarks demonstrate the model behavior under several training strategies such as MSMatch, Supervised Unweighted, and Supervised Weighted. The analysis, grounded in the MCC values, has revealed a significant performance of the method.

A breadboard of the E2E prototype was implemented by leveraging variegated hardware processing chains. The analysis of power consumption and latency of different hardware setups provides valuable practical insights highlighting how real-time performance is achievable by incorporating additional hardware like NCS2 and CogniSat-XE2, in contrast to the baseline Raspberry Pi 4 setup. The detailed examination of power and timing metrics across various configurations showcases the efficacy of these setups in reducing computational time and highlights the trade-offs in power consumption. These insights are integral to optimizing computational resources in real-time data processing scenarios, paving the way for more efficient and effective deployment of such systems in practical applications.

In summary, the crucial finding reported in the study demonstrates the feasibility of timely data processing in the realm of the constrained environment of onboard computing, marking a significant advancement of ML in space.

Acknowledgements

The authors would like to thank the ESA Advanced Concepts Team for providing access to their computing infrastructure that was instrumental in training the Machine Learning models used in this work.

References

[1] Chien, S., Doubleday, J., Thompson, D. R., Wagstaff, K. L., Bellardo, J., Francis, C., Baumgarten, E., Williams, A., Yee, E., Stanton, E., *et al.* Onboard autonomy on the intelligent payload experiment cubesatmission. *Journal of Aerospace Information Systems*, **2017**, 14(6): 307–315.

[2] Chien, S., Sherwood, R., Tran, D., Cichy, B., Rabideau, G., Castano, R., Davies, A., Lee, R., Mandl, D., Frye, S., *et al.* The EO-1 autonomous science agent. In: Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems, **2004**, 1: 420–427.

[3] Giuffrida, G., Fanucci, L., Meoni, G., Batič, M., Buckley, L., Dunne, A., van Dijk, C., Esposito, M., Hefele, J., Vercruyssen, N., *et al.* The Φ-sat-1 mission: The first

on-board deep neural network demonstrator for satellite earth observation. *IEEE Transactions on Geoscience and Remote Sensing*, **2021**, 60: 5517414.

[4] Kacker, S., Meredith, A., Cahoy, K., Labreche, G. Machine learning image processing algorithms onboard OPS-SAT. In: Proceedings of the 36th Annual AIAA/USU Conference on Small Satellites, **2022**.

[5] Mateo-Garcia, G., Veitch-Michaelis, J., Smith, L., Oprea, S. V., Schumann, G., Gal, Y., Baydin, A. G, Backes, D. Towards global flood mapping onboard low cost satellites with machine learning. *Scientific Reports*, **2021**, 11(1): 7249.

[6] Diana, L., Xu, J., Fanucci, L. Oil spill identification from SAR images for low power embedded systems using CNN. *Remote Sensing*, **2021**, 13(18): 3606.

[7] Meoni, G., Del Prete, R., Serva, F., De Beusscher, A., Colin, O., Longépé, N. Unlocking the use of raw multispectral earth observation imagery for onboard artificial intelligence. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, **2024**, 17: 12521–12537.

[8] Del Rosso, M. P., Sebastianelli, A., Spiller, D., Mathieu, P. P., Ullo, S. L. On-board volcanic eruption detection through CNNs and satellite multispectral imagery. *Remote Sensing*, **2021**, 13(17): 3479.

[9] Růžička, V., Vaughan, A., De Martini, D., Fulton, J., Salvatelli, V., Bridges, C., Mateo-García, G., Zantedeschi, V. RaVÆn: Unsupervised change detection of extreme events using ML on-board satellites. *Scientific Reports*, **2022**, 12: 16939.

[10] Mateo-García, G., Aybar, C., Acciarini, G., Růžička, V., Meoni, G., Longépé, N., Gómez-Chova, L. Onboard cloud detection and atmospheric correction with deep learning emulators. In: Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, **2023**: 1875–1878.

[11] Guerrisi, G., Del Frate, F., Schiavon, G. Artificial intelligence based on-board image compression for the Φ-sat-2 mission. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, **2023**, 16: 8063–8075.

[12] Guerrisi, G., Del Frate, F., Schiavon, G. Convolutional autoencoder algorithm for on-board image compression. In: Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, **2022**: 151–154.

[13] Melega, N., Longepe, N., Marchese, V., Paskeviciute, A., Aragon, O., Babkina, I., Marin, A., Nalepa, J., Buckley, L., Guerrisi, G., *et al.* Implementation of the Φsat-2 on board image processing chain. In: Proceedings of the SPIE 12729, Sensors, Systems, and Next-Generation Satellites XXVII, **2023**: 127290Z.

[14] Danielsen, A. S., Johansen, T. A., Garrett, J. L. Self-organizing maps for clustering hyperspectral images on-board a CubeSat. *Remote Sensing*, **2021**, 13(20): 4174.

[15] Helber, P., Bischke, B., Dengel, A., Borth, D. 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*, **2019**, 12(7): 2217–2226.

[16] Rijlaarsdam, D., Hendrix, T., González, P. T. T., Velasco-Mata, A., Buckley, L., Miquel, J. P., Casaled, O. A., Dunne, A. The next era for Earth observation spacecraft: An overview of CogniSAT-6. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, **2024**, 18: 2450–2463.

[17] European Space Agency. Sentinel-2 products specification document. Available at https://sentinel.esa.int/documents/247904/685211/sentinel-2-products-specification-document

[18] Massimetti, F., Coppola, D., Laiolo, M., Valade, S., Cigolini, C., Ripepe, M. Volcanic hot-spot detection using SENTINEL-2: A comparison with MODIS–MIROVA thermal data series. *Remote Sensing*, **2020**, 12(5): 820.

[19] Hu, C. L., Zhang, X. Y., Xing, X. W., Gao, Q. An approach to detect gas flaring sites using sentinel-2 MSI and NOAA-20 VIIRS images. *International Journal of Applied Earth Observation and Geoinformation*, **2023**, 124: 103534.

[20] Liu, Y. X., Pu, Y. L., Hu, X. Y., Dong, Y. Z., Wu, W., Hu, C. M., Zhang, Y. Z., Wang, S. H. Global declines of offshore gas flaring inadequate to meet the 2030 goal. *Nature Sustainability*, **2023**, 6(9): 1095–1102.

[21] Liu, Y. X., Zhi, W. F., Xu, B. H., Xu, W. X., Wu, W. Detecting high-temperature anomalies from sentinel-2 MSI images. *ISPRS Journal of Photogrammetry and Remote Sensing*, **2021**, 177: 174–193.

[22] Faruolo, M., Falconieri, A., Genzano, N., Lacava, T., Marchese, F., Pergola, N. A daytime multisensor satellite system for global gas flaring monitoring. *IEEE Transactions on Geoscience and Remote Sensing*, **2022**, 60: 5001717.

[23] Wu, W., Liu, Y. X., Rogers, B. M., Xu, W. X., Dong, Y. Z., Lu, W. Y. Monitoring gas flaring in Texas using time-series sentinel-2 MSI and landsat-8 OLI images. *International Journal of Applied Earth Observation and Geoinformation*, **2022**, 114: 103075.

[24] Fanizza, V., Rijlaarsdam, D., González, P. T. T., Espinosa-Aranda, J. L. Transfer learning for on-orbit ship segmentation. In: *Computer Vision – ECCV 2022 Workshops. Lecture Notes in Computer Science, Vol. 13801*. Karlinsky, L., Michaeli, T., Nishino, K. Eds. Springer Cham, **2023**: 21–36.

[25] Sarlin, P. E., DeTone, D., Malisiewicz, T., Rabinovich, A. SuperGlue: Learning feature matching with graph neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, **2020**: 4937–4946.

[26] Mateo-Garcia, G., Veitch-Michaelis, J., Smith, L., Oprea, S. V., Schumann, G., Gal, Y., Baydin, A. G., Backes, D. Towards global flood mapping onboard low cost satellites with machine learning. *Scientific Reports*, **2021**, 11: 7249.

[27] Meoni, G., Märtens, M., Derksen, D., See, K., Lightheart, T., Sécher, A., Martin, A., Rijlaarsdam, D., Fanizza, V., Izzo, D. The OPS-SAT case: A data-centric competition for onboard satellite image classification. *Astrodynamics*, **2024**, 8(4): 507–528.

[28] Tan, M., Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In: Proceedings of the 36th International Conference on Machine Learning, **2019**: 6105–6114.

[29] Gómez, P., Meoni, G. MSMatch: Semisupervised multispectral scene classification with few labels. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, **2021**, 14: 11643–11654.

[30] Chicco, D., Jurman, G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, **2020**, 21(1): 6.

[31] Delgado, R., Tibau, X. A. Why Cohen's Kappa should be avoided as performance measure in classification. *PLoS One*, **2019**, 14(9): e0222916.

[32] Jayadeva, Pant, H., Sharma, M., Soman, S. Twin Neural Networks for the classification of large unbalanced datasets. *Neurocomputing*, **2019**, 343: 34–49.

[33] Deng, J., Dong, W., Socher, R., Li, L. J., Kai, L., Li, F. F. ImageNet: A large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, USA, **2009**: 248–255.

[34] Medina, I., Hernández-Gómez, J. J., Torres-San Miguel, C. R., Santiago, L., Couder-Castañeda, C. Prototype of a computer vision-based CubeSat detection system for laser communications. *International Journal of Aeronautical and Space Sciences*, **2021**, 22(3): 717–725.

[35] Coyle, S., Burk, R., Kedrowitsch, A. EECSat: CubeSat development. In: Proceedings of the Annual General Donald R. Keith Memorial Capstone Conference, **2020**: 108–113.

[36] Lindsay, C., Sit, E. Open-source flight computer platform for CubeSats. In: Proceedings of the 34th Annual Small Satellites Conference, **2020**.

[37] Furano, G., Meoni, G., Dunne, A., Moloney, D., Ferlet-Cavrois, V., Tavoularis, A., Byrne, J., Buckley, L., Psarakis, M., Voss, K. O., *et al.* Towards the use of artificial intelligence on the edge in space systems: Challenges and opportunities. *IEEE Aerospace and Electronic Systems Magazine*, **2020**, 35(12): 44–56.

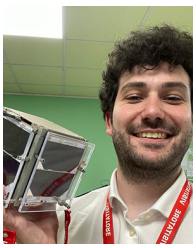
[38] Denby, B., Lucia, B. Orbital edge computing: Nanosatellite constellations as a new class of computer system. In: Proceedings of the 25th International Conference on Architectural Support for Programming Languages and Operating Systems, **2020**: 939–954.

[39] iTeh, Inc. Space engineering - SpaceFibre - Very high-speed serial link. CEN EN 16603–50–11:2020. **2020**.

[40] Meoni, G., Cassettari, R., Bertolucci, M., Marino, A., Davalle, D., Trafeli, M., Fanucci, L. CCSDS 131.2-B-1 telemetry transmitter: A VHDL IP core and a validation architecture on board RTG4 FPGA. *Acta Astronautica*, **2020**, 176: 484–493.

[41] Zabala, A., Vitulli, R., Pons, X. Impact of CCSDS-IDC and JPEG 2000 compression on image quality and classification. *Journal of Electrical and Computer Engineering*, **2012**, 2012(1): 761067.

Gabriele Meoni, received his Ms.C. degree in electronic engineering and his Ph.D. degree in information engineering from the University of Pisa respectively in 2016 and in 2020, where he was supervised by Prof. Luca Fanucci. After completing his doctoral studies, from September 2020 to April 2023 he held the position of internal research fellow at the European Space Agency (ESA) (Advanced Concepts Team (ACT) September 2020–August 2021, Φ-Lab September 2021–April 2023), where he conducted research on Artificial Intelligence (AI) and neuromorphic computing for onboard spacecraft applications. During 2022–2023, he was a visiting researcher at AI Sweden, focusing on distributed edge learning for satellite constellations. From May 2023 to April 2024, he served as an assistant professor in the Faculty of Aerospace Engineering at Delft University of Technology. Currently, Meoni is an innovation officer at ESA, with research interests spanning satellite onboard processing, AI for Earth observation, and neuromorphic computing. Meoni coauthored more than 40 scientific publications.



Roberto Del Prete, born in 1994, is an Italian researcher specializing in deep learning and edge computing for remote sensing. He focuses on improving time-critical decision-making through advanced AI solutions for space missions and Earth monitoring. He is pursuing a Ph.D. degree at the University of Naples Federico II, where he also earned his master and bachelor degrees in aerospace engineering. His notable work includes the

development of "FederNet", a terrain relative navigation system. Del Prete's professional experience includes roles as a visiting researcher at the European Space Agency's Φ-Lab and SmartSat CRC in Australia. He has received the 2022 NATO STO Early Career Scientist Award and participated in the 2021 NASA-ESA Trans-Atlantic Training. He has contributed to key projects like Kanyini Mission, and PYRAWS, developing AI algorithms for real-time maritime monitoring and thermal anomaly detection. With 28 scientific publications and 16 conference presentations, Del Prete is dedicated to leveraging advanced technologies to address global challenges in remote sensing and AI.

Lucía Ancos-Villa completed the degree of computer engineering, specializing in computing, at the University of Castilla-La Mancha, where she studied from 2020 to 2024. During her final years, she fulfilled an internship at Ubotica Technologies, contributing to the development of deep learning projects. Since 2024, she has been pursuing a master degree in applied artificial intelligence at Universidad Carlos III de Madrid. Her current academic and research focus lies in the field of image processing, employing advanced and innovative deep learning methodologies.

Enrique Albalate-Prieto is currently pursuing a master degree in artificial intelligence at the Universidad Politécnica de Madrid, a program he started in September 2024 and is set to complete in June 2025. He earned his bachelor degree in computer engineering from the Universidad de Castilla-La Mancha in June 2024, where he was honored with the award for the best academic record. He is currently working at Ubotica Technologies and his current research interests include theoretical machine learning, computer vision, and satellite onboard computing systems.

David Rijlaarsdam has a master of science in aerospace engineering from Delft University of Technology with a specialization in space system engineering. He is currently director of space system engineering for Ubotica Technologies, where he manages the Space System R&D team and is involved as a system engineer in several space missions. David has previously been part of the Automation and Robotics Section of the European Space Agency where he researched relative navigation for spacecraft and has been part of the advanced architecture team of Intel Movidius, where he researched the application of AI for star identification algorithms.

Jose Luis Espinosa-Aranda received his computer engineering degree and his Ph.D. degree in computer science from the University of Castilla-La Mancha, Spain, in 2009 and 2014, respectively. Currently, he works as director of the Spanish Headquarters of Ubotica Technologies managing system software integration and Ubotica's deep learning activities, while teaching as an associate professor at the University of Castilla-La Mancha, Spain. His work and research during the last 15 years has been focused on artificial intelligence, and his current interests are computer vision, machine learning methods and deep learning for embedded space applications.

Nicolas Longépé received his M.Eng. degree in electronics and communication systems and M.Sc. degree in electronics from the National Institute for the Applied Sciences, Rennes, France, in 2005 and his Ph.D. degree in signal processing and telecommunication from the University of Rennes I, Rennes, in 2008. From 2007 to 2010, he was with the Earth Observation Research Center, Japan Aerospace Exploration Agency, Tsukuba, Japan. From 2010 to 2020, he was with the Space Observation Division, Collecte Localization Satellites, Plouzané, France, where he was a research engineer. Since September 2020, he has been an Earth observation data scientist, Φ-Lab Explore Office, European Space Research Institute, European Space Agency, Frascati, Italy. His research interests include Earth observation remote sensing and digital technologies. such as machine (deep) learning. He has been working on the development of innovative synthetic aperture radar-based applications for environmental and natural resource management (ocean, mangrove, land and forest cover, soil moisture, snow cover, and permafrost) and maritime security (oil spills, sea ice, icebergs, and ship detection/tracking). At the Φ-Lab, he is particularly involved in the development of innovative Earth observation missions in which artificial intelligence is directly deployed at the edge (on the spacecraft).

Maria Daniela Graziano is an associate professor and lecturer of space systems and aerospace program management with the Department of Industrial Engineering of the University of Naples "Federico II". Her research interests include satellite constellation analysis and design, mission design for EO missions, distributed synthetic aperture radar (SAR), maritime sea traffic

monitoring by SAR data processing. She has been involved in several research projects funded by national and international institutions (ASI, ESA, Ministry of Defence) and her scientific production included more than 50 scientific papers published in distinguished international journals, conference proceedings, and book chapters, and frequently referenced in the literature.

Alfredo Renga received his M.S. degree (cum laude) in aerospace engineering and his Ph.D. degree in industrial engineering from the University of Naples Federico II, Naples, Italy, in 2006 and 2010, respectively. His research interests include earth observation from space, e.g., spaceborne synthetic aperture radar (SAR), maritime sea traffic monitoring, and autonomous navigation in planetary missions. He is an associate professor of aerospace systems with the School of Engineering, Naples, Italy. His scientific production includes more than 180 scientific papers (since 2007) published in distinguished international journals, conference proceedings, and book chapters, and frequently referenced in the literature. He held contracts and carried out research activities with the Second University of Naples, Naples, the University of Naples Parthenope, Naples, and CO.RI.S.T.A., Naples, a private research consortium on advanced remote sensing systems led by Thales Alenia Space Italy. He has been the scientific responsible of several research projects funded or supported by national and international institutions (ASI, DLR).