The OPS-SAT case

A data-centric competition for onboard satellite image classification

Meoni, Gabriele; Märtens, Marcus; Derksen, Dawa; See, Kenneth; Lightheart, Toby; Sécher, Anthony; Martin, Arnaud; Rijlaarsdam, David; Fanizza, Vincenzo; Izzo, Dario

DOI
10.1007/s42064-023-0196-y

Publication date
2024

Document Version
Final published version

Published in
Astrodynamics

The OPS-SAT case: A data-centric competition for onboard satellite image classification

Gabriele Meoni[1,2,3,*] (✉), Marcus Märtens[2,*], Dawa Derksen[2,3], Kenneth See[4], Toby Lightheart[4], Anthony Sécher[5], Arnaud Martin[5], David Rijlaarsdam[6], Vincenzo Fanizza[6], and Dario Izzo[2]

1. *Department of Space Engineering of the Faculty of Aerospace Engineering, TU Delft, Kluyverweg 1, 2629 HS Delft, the Netherlands*
2. *Advanced Concepts Team, European Space Agency, Keplerlaan 1, 2201 AZ Noordwijk, the Netherlands*
3. *Φ-lab, European Space Agency, Via Galileo Galilei 1, 00044, Frascati (RM), Italy*
4. *Inovor Technologies, SpaceLab Building, Lot Fourteen, Adelaide SA 5000, Australia*
5. *Capgemini Engineering – Hybrid Intelligence, 4-11 Avenue Didier Daurat, Blagnac, France*
6. *Ubotica Technologies, DCU Alpha, Old Finglas Road 11, Glasnevin, Dublin D11KXN4, Ireland*

ABSTRACT

While novel artificial intelligence and machine learning techniques are evolving and disrupting established terrestrial technologies at an unprecedented speed, their adaptation onboard satellites is seemingly lagging. A major hindrance in this regard is the need for high-quality annotated data for training such systems, which makes the development process of machine learning solutions costly, time-consuming, and inefficient. This paper presents "the OPS-SAT case", a novel data-centric competition that seeks to address these challenges. The powerful computational capabilities of the European Space Agency's OPS-SAT satellite are utilized to showcase the design of machine learning systems for space by using only the small amount of available labeled data, relying on the widely adopted and freely available open-source software. The generation of a suitable dataset, design and evaluation of a public data-centric competition, and results of an onboard experimental campaign by using the competition winners' machine learning model directly on OPS-SAT are detailed. The results indicate that adoption of open standards and deployment of advanced data augmentation techniques can retrieve meaningful onboard results comparatively quickly, simplifying and expediting an otherwise prolonged development period.

KEYWORDS

OPS-SAT
data-centric competition
artificial intelligence (AI)
onboard machine learning
onboard classification

Research Article

Received: 25 October 2023
Accepted: 21 December 2023

1 Introduction

Over the last decade, the potential demonstrated by artificial intelligence (AI) for the Earth observation (EO) has prompted the space community to investigate its suitability directly on board EO satellites for applications having strict requirements in terms of latency and downlink bandwidth [1–9]. In most previous studies, the training of machine learning (ML) models was possible only after the creation of suitable datasets for the target application.

Depending on the required level of quality, creating such datasets can be costly, involving the design of data-acquisition strategies, substantial preprocessing, and manual labeling of large quantities of data [2, 3, 6]. This issue is exacerbated when data produced by the mission imagers during operation are unavailable, as has been frequently reported when novel image sensors that have never been used in previous missions are deployed [1–3]. For some missions, such as Φ-Sat-1, the pixel signal-to-noise ratio and spatial resolution of Sentinel-2 data were manipulated to emulate the onboard sensor [2] in an attempt to mitigate the issue. However, performing

* Gabriele Meoni and Marcus Märtens contributed equally to this work.
✉ G.Meoni@tudelft.nl

Nomenclature

AI	artificial intelligence	ML	machine learning
EO	Earth observation	ODSET	original dataset
ESA	European Space Agency	PDSET	private dataset
FPGA	field programmable gate array		

such domain adaptations might add to the complexity of the data collection and training processes.

In the cases wherein such techniques prove inadequate, there is often no alternative to train the ML model after deployment. For instance, the *WorldFloods* model used in the WildRide mission [3] required to be retrained on the miniaturized RGB camera data, which were collected and labeled after the deployment of the mission in order to achieve acceptable performance. For sufficiently complex and specific applications, the deployment of such pipelines for collecting, post-processing, and labeling of data significantly increases the development time of the mission, impedes the reconfigurability of the satellite, and effectively shortens the mission lifetime [10]. Consequently, there is a significant need to accelerate this procedure or find shortcuts that enable ML models to be deployed as fast as possible.

To contribute towards this goal, this study investigated the following question: *How can a fixed ML model be effectively trained offline for onboard satellite classification if only a very reduced number of labeled data from EO satellites are available*?

The key aspects of this investigation are thus:

- **Very reduced availability of labeled data from EO satellites**: Limiting the requirements in terms of availability of labeled data from EO satellites for training should significantly accelerate the phase of dataset preparation. Sebastianelli *et al.* [11] demonstrated that dataset collection and preparation represent the most time-consuming design steps to train ML models for satellite imagery. Furthermore, labeling data for a specific application often requires a pool of application experts [12, 13]; this could significantly increase the cost and time for labeling.
- **Fixed ML model**: Modern ML models (e.g., deep neural networks) derive part of their performance from carefully engineered layers and operations that can be tailored to the task at hand. However, when operational constraints such as power consumption, bandwidth, and memory [1] need to be considered, a risk that overly specialized models might not be supported on board arises. Fixing the ML model architecture during the design of the mission minimizes such risks. A strict implementation of such an approach would prohibit the use of arbitrary and specialized ML models in favor of a single general ML model, whose behavior may only be reprogrammed by changing its parameters (e.g., weights and biases). Leveraging a fixed ML model would thus reduce the mission design time.
- **Raw satellite data**: Typically, "raw data" refer to the data directly produced by a satellite sensor, for example, the unprocessed images of an onboard camera. Such images are affected by different distorting phenomena, including radiometric distortion (e.g., the presence of additional noise, lack of sensor calibration, and stripe noise), geometric effects (due to the Earth's rotation, attitude disturbances, and other phenomena), and atmospheric distortion (e.g., Mie scattering, which distorts the image colors) [14]. Correction for such phenomena requires the development and automatization of extensive image processing pipelines. These are generally not well-suited for onboard application as their high computational cost adds to the time that is required to classify one image [15]. In contrast, by directly using raw data, image processing becomes part of the training of the ML model, further decreasing in the mission design and onboard processing time.

Considering each of these key aspects, we designed and conducted a data-centric competition called "the OPS-SAT case", leveraging mostly unlabeled and raw data produced by the OPS-SAT satellite. A basic land-cover classification problem was posed to resemble a common application task for EO satellites. Furthermore, a suitable ML model for this task was devised, and its application was made a strict requirement for the competition, accompanied by a rigid procedure for raw image classification. The onboard capabilities of OPS-SAT were closely emulated, and the competitors were enabled to train this ML model offline by using their

own methods. Thus, the competition design focused on conceiving effective training strategies and data augmentation rather than model architecture search, which is a typical aspect of ML competitions. We transferred the model of the winning team without any additional training steps to the satellite for onboard inference so as to test its performance directly in space.

The competition was conducted on the ESA's online platform Kelvins[①], which has hosted several challenges including the "Pose Estimation Challenge" [16, 17], "PROBA-V Super Resolution" [18], and "SpotGeo Challenge" [19].

This paper is structured as follows. Section 2 and Section 3 present related background for our target system, the OPS-SAT satellite, and a general overview of our vision for the data-centric competition. Section 4 details the dataset preparation for the competition, while Section 5 presents further details concerning the evaluation metrics, ML model selection, generation of baseline solutions, and overall evaluation process. Section 6 summarizes the results of the competition, including a description of the training and data augmentation procedures applied by the top 3 teams. Section 7 describes the results from the post-competition experiment in space, conducted during a flight campaign on board OPS-SAT by using the winning ML model. We summarize and discuss the results in Section 8, concluding with the insights gained concerning possible advancements in the development of onboard ML models in Section 9.

2 The OPS-SAT satellite

In 2019, the European Space Agency (ESA) launched OPS-SAT [20], a 3U CubeSat to orbit the Earth at an altitude of 515 km. OPS-SAT was designed as a technology demonstrator and space laboratory focusing on high-performance computation, resembling an advanced spacecraft, thus paving the way for future systems of ESA. The satellite is equipped with an Altera Cyclone V system-on-chip with an ARM dual-core Cortex-A9 MPCore and a Cyclone V field programmable gate array (FPGA), providing unprecedented capabilities for onboard software execution [20]. Instead of proprietary operating systems and specialized software commonly deployed in space, OPS-SAT aims to explore how general-purpose open-source software running on modern office personal computers or smartphones (e.g., Linux, Java, and Python) can be utilized to control the satellite itself.

ESA provides access to the OPS-SAT platform without cost and minimal bureaucratic overhead to European academia, industry, and interested individuals to schedule and run innovative experiments. Examples of such onboard experiments include the first execution of neuromorphic algorithms (based on spiking neural networks) in space [21], a fully autonomous planning and scheduling agent for image acquisition [22], and various demonstrations of onboard ML models [23].

In particular, the capabilities of OPS-SAT to execute an interpreter of TensorFlow Lite models, a popular and widely adopted industry standard for ML on edge devices [24], is essential for our research.

OPS-SAT can be interfaced from ESA's ground station via various means. The onboard S-band link allows for uplink speeds of up to 256 kbit/s and downlink speeds of up to 1 Mbit/s, while the X-band transmitter ensures a downlink data-rate of up to 50 Mbit/s. Considering all operational constraints, OPS-SAT can receive and execute TensorFlow Lite models of a maximum size of 10 MB. In addition to its ML capabilities, another essential feature of OPS-SAT is its optical camera (BST IMS-100), which serves as the source for the generation of our dataset. This camera can provide images and video with a ground resolution of up to 80 m × 80 m ground sampling distance per pixel. The attitude determination and control systems of OPS-SAT facilitate a high pointing accuracy of well below 1°. The images captured by OPS-SAT are received as portable network graphics (.png) with a resolution of at most 1944 × 2048. Figure 1 shows an example capture of the optical camera featuring a land scene with several atmospheric distortions, which we refer to as the "raw image", and its processed counterpart.

3 The OPS-SAT case: A data-centric competition

Public competitions have been proven effective in transferring state-of-the-art ML techniques toward challenging space applications, such as satellite image super-resolution [18], spacecraft pose estimation [25], or collision avoidance [26]. These competitions are typically designed around a split of a well-curated, space-related

① You can access ESA's Kelvins platform through https://kelvins.esa.int/

dataset: a large set of labeled and processed data is provided to the competitors to design and train ML models to the best of their capabilities. A more minor split from this data is provided without labels and serves as the test set. Following their submission to the competition platform, the competitor's models are tested on this test set and get ranked according to certain merit function, for example, a mean-square error between the inferred results and the hidden labels of the test set. Such a competition design incentivizes the development of powerful but often also prohibitively large and sophisticated ML models and can thus be seen as "model"-centric or "algorithm"-centric competitions.

For the OPS-SAT case, whose concept is illustrated in Fig. 2, we applied an alternative competition design. We refer to this design as "data-centric" competition. It differs from the aforementioned designs in the following aspects:

(1) The majority of the training set consists of unlabeled and unprocessed patches, while the labeled examples

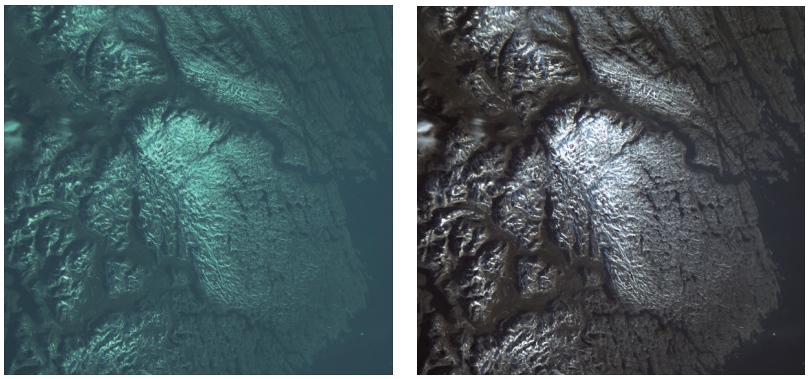


Fig. 1 Left: raw capture of a ground-scene from the OPS-SAT optical camera. Right: an offline processed version of the same image.

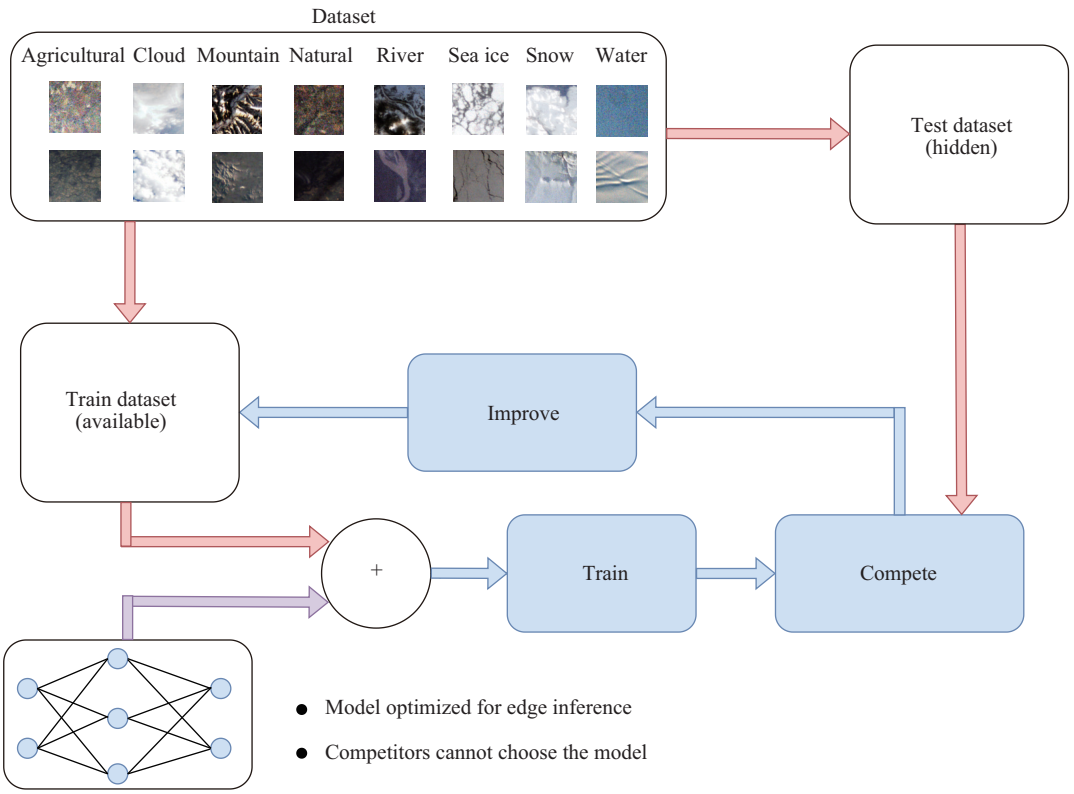


Fig. 2 Concept of "the OPS-SAT case" competition. Patches of the dataset are shown post-processed for visual aid.

are comparatively an exiguous minority, with only a few labels serving as examples.

(2) The test set is completely hidden from the competitors, preventing them from running inference on it directly.

(3) The ML model is fixed and cannot be freely chosen.

By adhering to these design principles, the competitors can train the given ML model by using the provided and additional data by using their own conceived training strategy. The inference on the test set is then performed by the organizers by instantiating the ML model and loading the submitted parameters, eventually obtaining a ranking of all submissions to determine the winning team.

Following this "data-centric" design, the OPS-SAT case competition is centered around a classification task concerning patches of the raw images provided by OPS-SAT. A few examples of land-cover types, such as Mountain, Snow, Cloud, Sea ice, Water, and others, are provided with many unlabeled raw images. Competitors shall find and submit the parameters of a fixed ML model without access to the test set's image patches.

Imposing these limitations on the competitors allows for the faithful replication of the onboard algorithmic procedures of OPS-SAT, facilitating the transfer of the results of the competition to the satellite without additional steps involved. Denying the competitors access to the test set patches for inference and forcing them to work with unprocessed raw images for training align with the specific difficulties of the onboard application domain, emulating a live acquisition of previously unseen images directly by the OPS-SAT camera.

4 The OPS-SAT case dataset

The creation of the OPS-SAT case dataset entailed several steps, as shown in Fig. 3. Beginning with raw captures from the OPS-SAT camera, this procedure aimed to create numerous small and labeled 200 px × 200 px patches from the larger 1944 px × 2048 px OPS-SAT images, removing defects and selecting mountains, rivers, clouds, and other distinguishable features. An image processing step was needed to compensate for the effects of atmospheric distortion and aid our team of experts in the labeling process. As our labeling campaign resulted in a non-negligible imbalance among target classes, additional measures were devised to create a meaningful split between the training and test part of the dataset, with the hidden test part containing a significant amount of labeled patches. Meanwhile, the training part is composed of raw images with only a few labeled patches per target class as examples. The training part of this dataset was released on Zenodo [27] at the beginning of the OPS-SAT case competition. In contrast, the test part was released after the end of the challenge [28]. This section details each of the above mentioned steps, highlighting several technical and practical challenges that are emblematic of ML tasks related to EO.

4.1 Edge image removal

Generally, OPS-SAT images can be considered to belong to one of two categories: *Edge* or *Earth*, as defined by Labrèche *et al.* [22]. Edge images suffer from a high elevation angle, containing part Earth and part sky, whereas Earth images are close to the nadir (see Fig. 4 for an example). Such Edge images occur frequently in the context of OPS-SAT, as the satellite was affected by a tumbling motion during the dataset acquisition.

Although we initially considered classifying images belonging to either Edge or Earth part of the competition, we adopted the more challenging land-cover classification task instead.

The removal of the Edge images from the dataset was

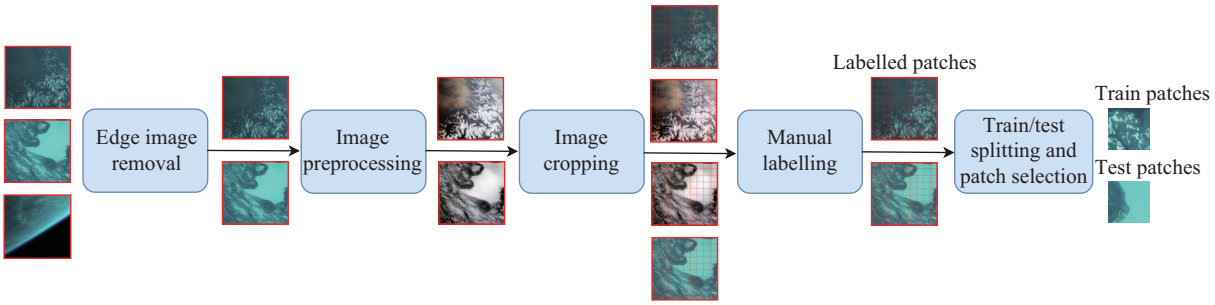


Fig. 3 Dataset creation procedure.

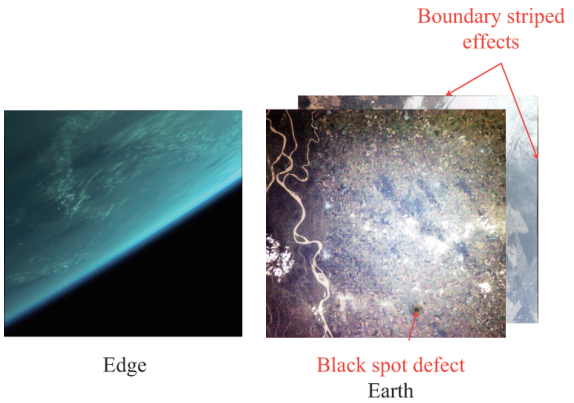


Fig. 4 Edge (unprocessed) and Earth images (white-balanced). Boundary striped and back spot defects are shown on white-balanced Earth images.

performed using an approach similar to that described in Section 5.3. Inspecting the results of our Edge images removal procedure showed a near perfect accuracy; therefore, the given problem is already easily solvable on board. Consequently, only Earth images were considered for the remainder of this study and in the OPS-SAT case competition.

4.2 Image preprocessing

Raw OPS-SAT images are not calibrated or corrected for atmospheric absorption; thus, their accurate labeling is challenging even for experts. Therefore, we applied successive steps of contrast enhancement and histogram equalization, taken off the shelf from the EO image processing toolkit *Orfeo ToolBox* [29]. Although this procedure enhanced the noise in certain areas, the overall visual quality of the images improved, paving the way for the manual labeling procedure. We emphasize that the patches of the enhanced images are neither part of the public training nor the hidden test set but were only deployed as a tool for obtaining human labels.

4.3 Image cropping

After surveying all the Earth images captured by OPS-SAT, we selected a small subset containing a diverse set of land-cover features for the image cropping step. In this step, we divide each OPS-SAT image into patches of size 200 px × 200 px each to match the onboard memory requirements and to isolate specific features. We extracted the patches from the original images without overlap by following a regular grid.

Before extracting the patches, we removed 10 pixels from the boundary of each image because a sensor defect produced striped noise in those areas. In addition, we removed one of the patches due to a different sensor defect from each image, resulting in a dark spot on fixed coordinates. Both of these defects are shown in the processed Earth image of Fig. 4. We produced a total of 1941 patches to serve as a starting point for the labeling procedure, making each patch's raw and processed version available to our labelers.

4.4 Manual labeling

Upon studying our set of patches, we initially defined nine classes of interest: Agricultural, Cloud, Desert, Mountain, Natural, River, Sea ice, Snow, and Water. All of these classes are representative of geological features or, in the case of Cloud, atmospheric features that can be directly discerned from the color and texture of the corresponding patch. Defining classes with distinct visual features (i.e., the meandering pattern of a river or the white color of snow) was necessary to guide our group of expert labelers in their decision process, as no geo-reference was obtained for the OPS-SAT images during acquisition. In particular, we defined Natural patches as rural, not-exploited lands that are not mountains or deserts and do not include rivers or traces of human activities. The feature that distinguishes Natural and Agricultural patches is the absence/presence of human-cultivated crops. In the case of snowy mountains, the labelers were asked to label as "Snow" those patches containing more than 50% of snow pixels.

However, patches lacking distinguishing features, including artifacts or enhanced noise due to the image preprocessing step, still exhibited significant ambiguity. For example, patches containing water without coastal areas were often indistinguishable from desert patches, especially when their colors appeared significantly distorted. This difficulty, however, could be mitigated by contextualizing the patch within the entire uncropped image to enable the labeler to grab additional information from the surrounding patches. To this end, we equipped the labelers with complete, uncropped satellite images.

We enforced a strict voter consensus to avoid including ambiguous patches in the dataset. More precisely, all patches with a voter consensus of less than 6 out of 8 votes were discarded. Checking the discarded patches after the labeling campaign confirmed that these contained either blurry or noisy visuals or featured overlapping classes,

Table 1 Labeling statistics. The retention rate is the ratio of samples that have a high voter agreement (more than 6 out of 8 votes) to the total number of patches that have at least one vote for the class. It shows how often a high number of voters agreed on a certain class label

	Retention rate	Patches in final test set
Mountain	46.9%	99
Cloud	54.5%	114
Snow	58.4%	106
Water	60.8%	107
Agricultural	69.0%	36
Sea ice	84.6%	37
Desert	43.4%	0
Natural	27.5%	58
River	48.2%	31

inducing ambiguity. Table 1 summarizes the dataset's retention rate and final number of patches per class.

4.5 Train/test split and patch selection

To prevent data leakage between the training and test sets, which could result in underestimating the true classification error, we made the train/test splitting at image level. In other words, all the patches cropped from the same image were either included as part of the training or the test set but never distributed between them. In particular, the training set was populated by using 10 labeled patches per class by selecting those with perfect agreement among the labelers. Furthermore, we provided the 26 raw satellite images to be used by the competitors to procure unlabeled data.

At this point, our general competition design had progressed already so far that it was certain that we would use the EfficientNet-lite0 neural network for the ML model of our choice (discussed in more detail in Section 5.1). We trained EfficientNet-lite0 on several different dataset splits and measured its performance not only as a proof of concept but also to select a dataset split of good quality. The training procedure used for these experiments is the same that was use to create the competition baseline, as outlined in Section 5.3.

Figure 5 shows the confusion matrix of the trained EfficientNet-lite0 for one of our iterations. The most remarkable challenge for the model was to distinguish the Mountain from the Snow class. This was expected, as features from the Snow and Mountain classes appeared commonly in the same patch, making this particular case ambiguous. We found that this type of ambiguity would make for an interesting challenge without compromising

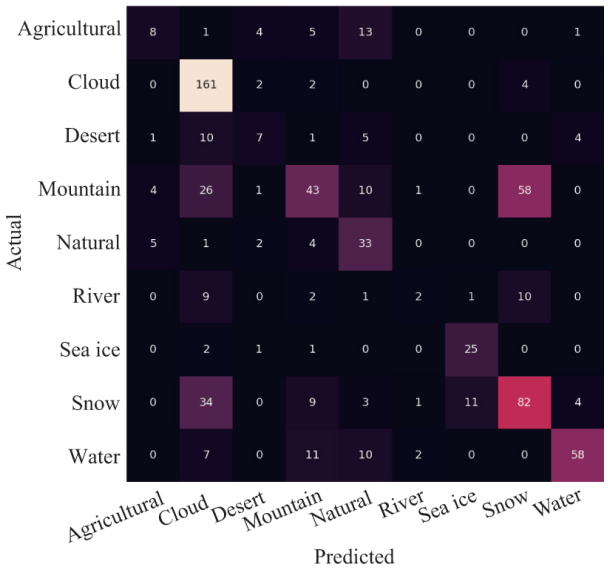


Fig. 5 Confusion matrix of the trained EfficientNet-lite0 model on an intermediate dataset splitting iteration.

the quality of the competition dataset. Therefore, we decided to keep both the Mountain and Snow classes in the final dataset.

Despite the poor consensus rate (27%) among the labelers, the Natural class was surprisingly well classified by the trained EfficientNet-lite0 model. This indicates that the consensus filtering retained few but valid and recognizable patches for training. In contrast, the Desert class featured a low number of test samples, indicating rarity in the dataset, combined with the difficulty faced by our human labelers to correctly classify these patches. Simultaneously, the EfficientNet-lite0 model showed low accuracy when confronted with these patches. Therefore, this class was removed from the final dataset. Although the trained EfficientNet-lite0 was nearly incapable of classifying the River class, we decided to include it in the final dataset to provide more variety and challenge for the competitors.

In summary, the final dataset version includes 8 classes corresponding to the initial class selection except for Desert. In particular, the final training subset contains 80 labeled patches and 26 raw images for the public training set. Notably, the number of unlabeled patches available to the competitors depends on the algorithm used to crop the 26 original images. The algorithm was never released so as to allow the participants to design their own dataset creation strategy. The hidden test set consists of 588 labeled patches. The number of patches for each class in the final test set is presented in Table 1.

5 Competition design

5.1 Choice of the ML model

To design the model for the competition, we considered both the classification performance and onboard satellite requirements. Nanosatellites such as CubeSats usually feature strong limitations in terms of power-budget, uplink, and downlink bandwidth [1]. In our specific case, the main onboard limitation was due to the uplink bandwidth, restricting the size of the ML model to 10 MB.

At the time of the competition design, EfficientNet models [30] offered the best trade-offs in terms of model size and classification performance. In particular, we selected the EfficientNet-lite0 architecture, a modified version of the EfficientNet-B0 suitable for embedded systems. Owing to the removal of squeeze-and-excitation networks and replacement of all the activations with RELU6, EfficientNet-lite0 supports post-training quantization, resulting in a significant shrinking of the model size [31].

To test the suitability of the EfficientNet-lite0 model on the OPS-SAT case dataset, we created a baseline solution as described in Section 5.3. Application of post-training quantization to the original 32-bit floating-point model, which transformed all parameters into the 16-bit floating-point format, resulted in a negligible loss in performance while reducing the model size down to 5.6 MB. We did not face any underflow or overflow on the test subset after halving the model precision. Consequently, we decided on the EfficientNet-lite0 model as the fixed ML model to be used for the competition, with the post-training quantization as the fixed step performed on every submission as part of their evaluation.

5.2 Evaluation metrics

As detailed in Section 4, the test dataset is imbalanced with classes such as Agricultural, River, or Sea ice appearing with lower frequency than classes such as Snow, Water, or Clouds. Deploying accuracy as the evaluation metric for ranking submitted models during the competition would induce a bias towards over-represented classes, with errors in less frequent classes having only negligible impact. Thus, instead of accuracy, we decided to select our own metric $\mathcal{L}$ (Eq. (1)) for this competition:

$$\mathcal{L} = 1 - \kappa \tag{1}$$

where κ is *Cohen's kappa* coefficient. The *Cohen's kappa* has already found extensive application as a metric for imbalanced classification problems for various remote sensing applications [32–34]. The value κ represents a measure of agreement between two different raters, which, in our case, would be the labels of the test set and predictions of the ML model. In the case of perfect agreement, $\kappa = 1$, while $\kappa = 0$ if classes are predicted at random according to their relative frequency. To be consistent with ESA's competition platform Kelvins, the definition of our metric $\mathcal{L}$ switches this meaning, making $\mathcal{L} = 0$ the hypothetically best achievable score[②] and $\mathcal{L} = 1$ corresponding to a score obtained by a random assignment.

Following the requirement to apply a 16-bit floating-point quantization, we define $\mathcal{L}_{\text{quant}}$ as the $\mathcal{L}$ value computed after applying quantization to the submitted network parameters. The leaderboard is ranked according to $\mathcal{L}_{\text{quant}}$. Meanwhile, the values of unquantized score (i.e., $\mathcal{L}$) were collected to investigate the influence of the post-training quantization with regard to the score.

5.3 Competition baseline

We based our training pipeline on a variant of *FixMatch* [35], called *MSMatch* [13], a semi-supervised training pipeline that has been developed for remote sensing applications. MSMatch is based on pseudo-labeling and consistency regularization to create an additional loss to the *categorical cross-entropy*, generally used for supervised training for multi-class classification problems. In particular, strong and weak augmentation are applied to an unlabeled image. Then, if the model's prediction on the weakly augmented image passes a certain confidence threshold, a consistency regularization loss is created to teach the model to predict the same class for weakly and strongly augmented images.

Compared to the original implementation that utilizes a seeded random sampling of the entire dataset to create the training and test datasets, we used the split created via the procedure described in Section 4.5. In particular, we used the selected ten examples per class as labeled part of the dataset. To create an unlabeled portion, we cropped the full images of the train set provided to the competitors into 200 × 200 patches and used them as

② Challenges on ESA's Kelvins platform are traditionally designed to reach an "absolute zero" as a score, requiring the score to be always minimized.

unlabeled data. Furthermore, in contrast with the original *MSMatch* implementation, we did not apply any image normalization as it would require us to utilize undisclosed channel statistics, such as mean and standard deviation, of the test set.

To train the baseline model, we used a batch size of 16 labeled patches, an unlabeled ratio of 7 (which is the number of unlabeled patches for each labeled one), a weight decay of 0.00075, a learning rate of 0.03, and a pseudo-label confidence threshold of 0.95. All training was performed using PyTorch [36], followed by a conversion to the TensorFlow Lite format. After 400 epochs of training, the baseline achieved a score of $\mathcal{L}_{\text{quant}} = 0.539694$ and 46.22% accuracy.

5.4 Submission evaluation

For the success of this competition, it was crucial to prevent the participating teams from probing the hidden test set and exploiting the public leaderboard to gather excessive information about it. Consequently, we implemented the following measures:

- Each team could score at most two submissions per day.
- Only the score of the best submission for each team was made public.
- Submissions were evaluated on only 50% of the hidden test set during the submission period.

Figure 6 shows the evaluation and ranking procedure regarding a single submission.

We forced the participants to submit a single.h5 file containing a trained EfficientNet-lite0 model, whose architecture was provided in the open code repository of the OPS-SAT case starter-kit [37]. More precisely, we derived the final implementation of the competition model by removing the "stem_activation" layer from the original Keras implementation [38] provided by *efficientnet-lite-keras*[③] to facilitate the conversions from the PyTorch users relying on *efficientnet-lite0-pytorch*[④].

We discarded a submission if the model submitted did not match the one provided in the starter-kit. Otherwise, the submitted model was converted into a TensorFlow Lite model and quantized into a 16-bit floating-point format to match the uplink bandwidth requirements of the OPS-SAT satellite, as detailed in Section 5.1. During the submission period of the competition, we performed the inference on a fixed 50% subset of the test set, with both metrics $\mathcal{L}$ and $\mathcal{L}_{\text{quant}}$ as a result. Only if the team improved upon their previous score (or had no previous submission) was an update on their rank in the leaderboard, according to $\mathcal{L}_{\text{quant}}$, reported.

To support the participants on a technical level, we made the serverside evaluation code, model conversion utilities, and helper functions for generating valid submission files available in the open code repository of the OPS-SAT case starter-kit.

6 Competition results

6.1 Competition outcome

We extended invitations to expert teams and individuals worldwide to participate in the competition. Overall, 56 teams registered, of which 41 managed to enter the leaderboard by producing at least one valid submission. Submissions were possible during a four-month period, ranging from July 1 to October 31, 2022.

Figure 7 shows a timeline for the score ($\mathcal{L}_{\text{quant}}$) during the submission period. A total of 891 valid submissions were received by the end of the competition. The best submission of each team of the public leaderboard was re-evaluated on 100% of the hidden test set to determine the final ranking. This re-evaluation did not change the ranking of the top 3 teams, indicating no overfitting. The final results of the 48 teams produced a $\mathcal{L}_{\text{quant}}$ ranging from 1.002 (worst result) to 0.381 (competition winner).

The baseline solution with $\mathcal{L}_{\text{quant}} = 0.5397$ would have been ranked the 11th place in the leaderboard, if it was submitted. Table 2 shows the top 10 teams of the final leaderboard. Although the final ranking was determined on the basis of $\mathcal{L}_{\text{quant}}$, the impact of the quantization error was insignificant as it did not affect the ranking of the top teams. In contrast, adopting accuracy as the ranking metric would have significantly impacted the ranking, in particular, moving the 4th ranked team Alcheringa-Dreamtime up to the 3rd place.

6.2 Methods of the top 3 winning teams

We interviewed the top 3 teams to examine the methodologies that led to their respective results. It was confirmed that the scarcity of labels was forcing the competitors to invest most of their efforts into creating their own version of the data for training the model, which

③ https://github.com/sebastian-sz/efficientnet-lite-keras
④ https://pypi.org/project/efficientnet-lite0-pytorch-model/

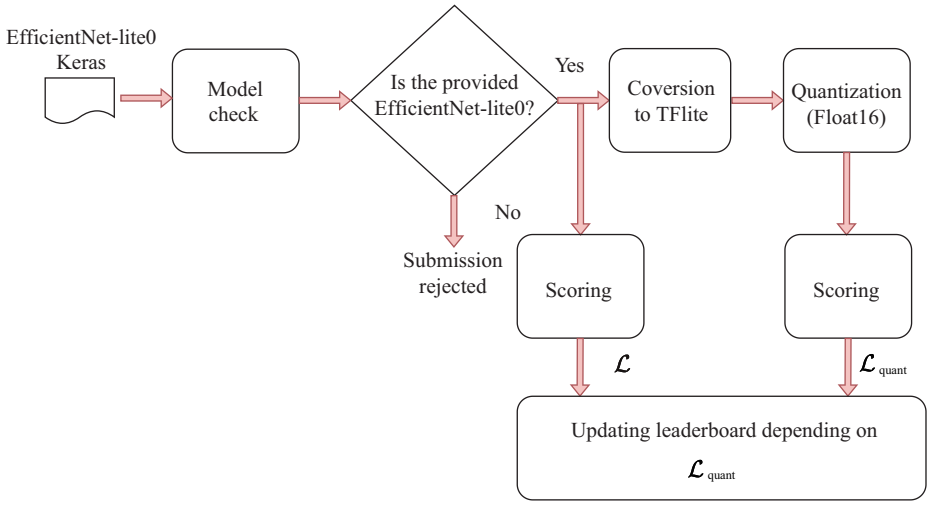


Fig. 6 Submission workflow.

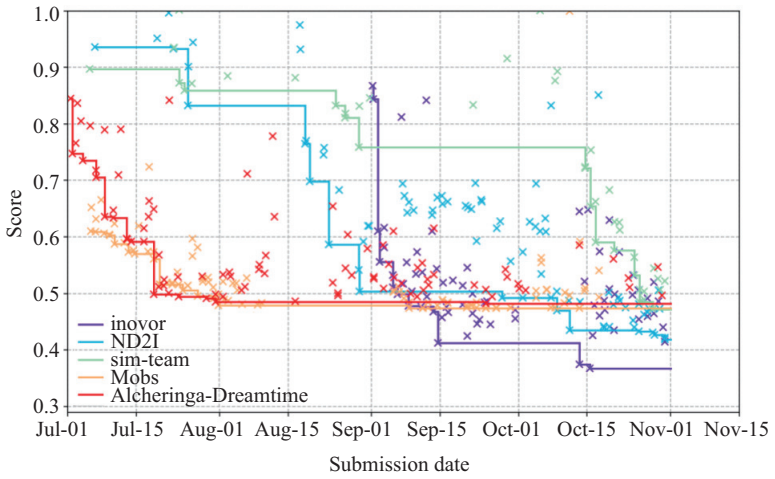


Fig. 7 Evolution of public leaderboard score for the top 5 teams. Each cross corresponds to a valid submission. Solid lines indicate the best submission (lower values are better).

Table 2 Final ranking of top 10 teams evaluated on the whole dataset

Rank	Team name	$\mathcal{L}$	$\mathcal{L}_{\text{quant}}$	Accuracy
1	inovor	0.38109	0.38109	0.67857
2	ND2I	0.40868	0.40870	0.65646
3	sim-team	0.47429	0.47429	0.59694
4	Alcheringa-Dreamtime	0.47635	0.47618	0.60034
5	Mobs	0.49790	0.49001	0.58503
6	deya109	0.50317	0.50114	0.57823
7	AgeniumTeam	0.50456	0.50456	0.57143
8	DOTE_GTDxIRT	0.50624	0.50812	0.56463
9	vision_creation	0.51393	0.51355	0.55952
10	perico	0.52893	0.52880	0.55272

was precisely our intention as organizers of this data-centric competition. To allow for better differentiation herein, we refer to the original, sparsely labeled OPS-SAT case dataset as the "original dataset" (abbreviated: ODSET). Any augmented or modified version of the data by the competitors is referred to as a "private dataset" (abbreviated: PDSET), as those were the keys for success and thus never publicly shared by any team.

In addition to the interviews, deploying the submitted models to the ODSET allowed us to analyze their performances on a finer level than on the basis of the scoring metric alone. Figure 8 shows the confusion matrices of the best submission for each team evaluated on the hidden test set of the ODSET. Figure 9 compares the patch classification on an image from the training set of the ODSET. A summary of the different methodologies of the top 3 teams is presented in Table 3. The remainder of this section presents the details concerning the training

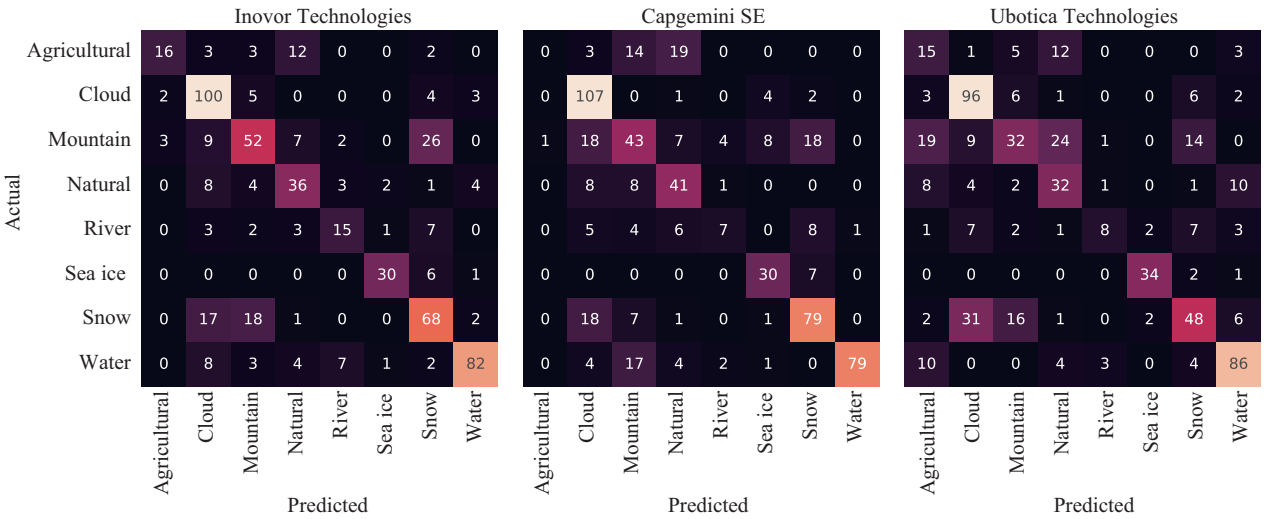


Fig. 8 Confusion matrices of the best submissions of the top 3 winning teams.

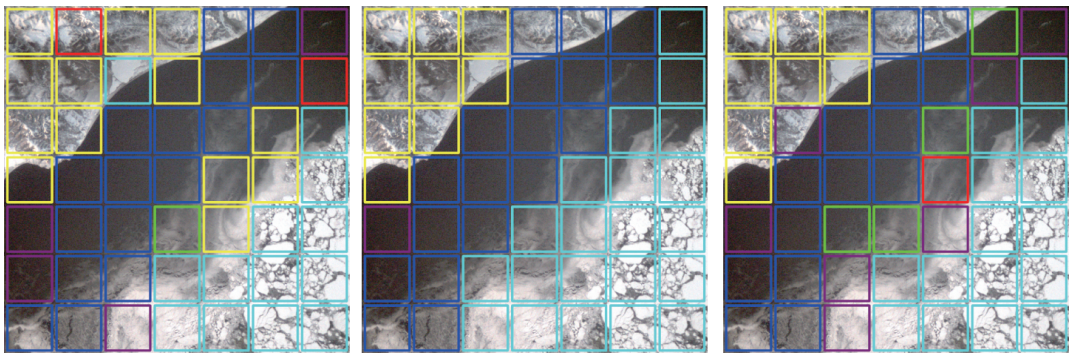


Fig. 9 Classification of an unlabeled image from the official OPS-SAT case dataset based on the best submission of the top 3 teams. From left to right: Inovor Technologies, Capgemini SE, Ubotica Technologies.

and data augmentations that the competitors chose to share with us.

6.2.1 Inovor Technologies (1st rank)

The PDSET of the competition winners was generated as follows. First, the unlabeled images of the ODSET were cropped into multiple patches. Second, a new source of OPS-SAT images was added by collecting the images from the official OPS-SAT *Flickr* album[⑤]. This online photo album was launched to showcase OPS-SAT images for public viewing and therefore contains post-processed images different from the raw images of the ODSET. To effectively utilize these images, Inovor Technologies applied linear scaling to match the color-channel statistics of the labeled patches of the ODSET. The pixel mean and standard deviation were computed separately for each class and applied to normalize the patches from the Flickr images after they received their label.

Patch labeling was performed manually by the team itself, generating the first submissions by training the model only on patches with a high confidence level. Exploiting the feedback of the public leaderboard, patches with lower confidence level were corrected and incrementally added to the training set once their quality could be assessed. For the final submission, the PDSET used for training was composed of approximately 75% patches from the normalized Flickr images.

Data augmentation was used to increase the size of the competitor's PDSET accordingly; this includes cropping, zooming, rotation, brightness, contrast, hue and saturation jitter, perspective shifting, and gamma corrections. Inovor Technologies based their augmentations on the assumption that the saliency of geometrical and color features ought to be different for each class, accordingly tuning the intensity of the various

⑤ https://www.flickr.com/photos/esa_events/albums/72157716491073681/

Table 3 Comparison of methodologies of the top 3 teams

	Inovor Technologies	Capgemini SE	Ubotica
Usage of additional datasets	OPS-SAT Flickr	NWPU-RESICS45 [39]	—
Manual labeling	Yes	Yes	Yes
Pseudo labeling	(Not disclosed)	Yes	Yes
Data augmentations	Cropping, zooming, rotation, brightness, contrast, hue, and saturation jitter, perspective shifting, and gamma corrections	Vertical and horizontal flip, random brightness and contrast, solarize, RGB shift, coarse dropout, shift scale rotate, and resize	Flips and rotations
Size of final dataset	(Not disclosed)	12,583	3000–7000
Optimizer	Adam	RMSprop	Adam
Loss function	Focal Loss	Sigmoid Focal Cross Entropy	Sparse Categorical Cross Entropy + Cosine Similarity
Model trained	EfficientNet-lite0	EfficientNet-lite0	Ensemble of EfficientNet-lite0 + Xception [40]

augmentations. For instance, as a more substantial presence of geometrical features characterizes patches from the Agricultural and Natural categories, applying stronger augmentation techniques for these classes was possible. In contrast, Mountain and Snow patches required a lower color augmentation as color was one of the most distinguishing features between these classes.

As regards their final submission, the ML model was trained using PyTorch for 80 epochs with Adam as optimizer and a focal loss function to handle dataset imbalance.

6.2.2 Capgemini SE (2nd rank)

Capgemini SE participated under the team name "ND2I" in the competition. The PDSET of this team was developed and grown three times during the submission period. First, the unlabeled ODSET images were cropped using a 40% overlap into patches. The most representative of these patches were manually labeled, creating a first set of 3679 patches.

Second, images from the external NWPU-RESICS45 dataset [39] were added. This dataset contains 31,500 images from various sources divided into 45 scene classes and has been published for free use to support research into remote sensing. In order to adapt patches from NWPU-RESICS45 images to better resemble the raw patches of the ODSET, a custom variant of the optimal transport method [41–44] was deployed. Considering the 80 labeled patches of the ODSET as a colorimetric reference, an optimal vector minimizing the transport cost was computed to replace the value of the source pixel with the value of the target pixel, transferring the color range of the ODSET to the NWPU-RESICS45 patches, artificially recreating their blueish hue. After this step, the size of the PDSET increased to 5578 labeled patches.

In a final step, Capgemini SE trained the EfficientNet-lite0 model with self-training on 520,000 patches cropped from the raw ODSET images using pseudo-labeling. The generated pseudo-labels were used to retrain a new instance of the model from scratch in an iterative process. During each iteration, training was performed for 5 epochs on the pseudo-labeled patches provided from the previous iteration and evaluated by using the 80 labeled patches of the ODSET. Capgemini SE manually corrected misclassified patches if the prediction confidence was greater than 0.95 in order to reduce the bias induced by the auto annotation. Furthermore, Capgemini SE visually inspected and double-checked pseudo-labels of underrepresented classes (e.g., River, Agricultural, or Sea ice) to correct possible errors. In total, 2808 of such patches were reviewed by a human for validation. Subsequently, the final size of Capgemini SE's PDSET reached 12,583 labeled patches.

Concerning the training strategy, the EfficientNet-lite0 model was pre-trained using the complete PDSET. Then, the weights originating from the pre-training were used to initialize a new instance of EfficientNet-lite0 that was trained leveraging only the 3679 labeled patches created during the first step of the PDSET. This allowed for the model to converge quicker to better solutions as compared to using random starting weights.

During the training, Capgemini SE also used several data augmentation techniques: vertical and horizontal flip, random brightness and contrast, solarize, RGB shift,

coarse dropout, shift scale rotate, and resize. These augmentations were provided using the AutoAlbument function from the Albumentations library [45].

The loss function used was a sigmoid focal cross entropy. This family of loss functions can be adapted to force an ML model to focus on classes of lesser frequency within the training set, making it particularly useful for situations with class imbalance [46, 47]. Specifically, Capgemini SE deployed a parameter of $\gamma = 3.1$ and a weighting according to the relative frequency of the classes in the PDSET:

$$\alpha = [n_{\text{class1}}/N_{\text{tot}}, n_{\text{class2}}/N_{\text{tot}}, \cdots, n_{\text{class8}}/N_{\text{tot}}] \quad (2)$$

Additionally, label smoothing was deployed to regularize the one-hot encoded labels according to the α factor using

$$y_k^{\text{LS}} = y_k(1-\alpha) + \alpha/K \quad (3)$$

where $K = 8$ is the number of classes. Smoothing the labels prevents the model from becoming over-confident by implicitly aligning its confidence with the accuracy of its predictions [48, 49].

Lastly, RMSprop was selected as the optimizer, softmax as the activation function, and 0.5 as the dropout rate. To limit overfitting, the callbacks used are EarlyStopper and ReduceLROnPlateau. The train/test split of the PDSET (85/15) was generated by using the stratify method to account for the class imbalance.

6.2.3 Ubotica (3rd rank)

Ubotica participated in the competition under the name "sim-team". The PDSET of this team was developed using manual labeling, semi-supervised learning, and data augmentations.

In the first step, Ubotica took the unlabeled images provided by the ODSET and labeled them manually on a pixel level, creating a partial image segmentation. These segmented images were then cropped into patches, allowing for an overlap of 100 px × 100 px at most. Only patches having at least 60% labeled pixels were retained, as only some regions could be confidently annotated. The resulting PDSET had a size of approximately 5000 labeled patches with a significant class imbalance.

In the next step, Ubotica increased the size of the PDSET by an iterative training and pseudo-labeling procedure, by using an EfficientNet-lite0 model trained on the manually labeled patches for 100 epochs, with sparse categorical cross entropy as loss function. To prevent the model from being biased by Ubotica's labels, this model was fine-tuned by retraining it for 20 epochs on the labeled patches of the ODSET. Cosine similarity was selected as loss function for this second training step, as it provides better performance for neural network training on small datasets [50]. Dropout rates were set to 0.6 and 0.4 for the first and second training steps, respectively. The Adam optimizer was used for both steps to train the network with a batch size of 8.

After data augmentation, the trained model was deployed to generate pseudo-labels for unlabeled patches cropped from the ODSET images. Ubotica worked under the assumption that the data distribution between the classes of the train and test part of the ODSET ought to be skewed. As the test set of the ODSET was not available to assess statistics to compensate for such skew, deploying strong data augmentation techniques such as adapting the patch's brightness, hue, or contrast was deemed potentially detrimental. Thus, only simple flips and rotations were deployed, leaving the domain of the data mostly unchanged. In particular, Ubotica used the flip and rotation transformation to create five versions of each unlabeled patch and included the pseudo-label into their PDSET only if the trained model agreed on all five patches with a confidence of at least 99%.

This pseudo-labeling procedure was used several times to augment the PDSET and fine-tune the parameters of the EfficientNet-lite0 model, evaluating its performance during experiments and making submissions to the public leaderboard. Depending on the iteration of this process, the PDSET contained approximately 3000–7000 labeled patches.

As regards the final submission of Ubotica, an ensemble of different neural networks was trained to create the final version of the PDSET and weights of the submitted EfficientNet-lite0 model. The idea behind deploying different model architectures for training was to allow for convergence to a better local minimum on the loss surface, thus increasing the robustness of the overall prediction of the ensemble. More precisely, a 3-ensemble was created by mixing one Xception network [40] with two distinct instances of an EfficientNet-lite0 model. The Xception network was trained using the same training and fine-tuning procedure as described for the EfficientNet-lite0 model. After training, the ensemble was used to create a new version of the PDSET as follows: all models were shown five different flips and rotations of the same patch,

and, only if all predictions agreed with high confidence, the pseudo-label of the patch was included into the PDSET. Seven training and fine-tuning iterations were needed to produce Ubotica's best performance.

7 Post-competition in-orbit experiment

Following the conclusion of the competition, we organized an experimental campaign to evaluate the performance of the trained models directly on board the OPS-SAT satellite. In particular, owing to its superior performance, the quantized model of Inovor Technologies was selected and uploaded to the satellite. During the experimental campaign, the satellite collected several raw images, which were cropped into 200 px × 200 px patches and directly fed into the model for inference by the satellite. The resulting activation of the last layer was saved in a log file, which was transferred back to the Earth alongside the raw images and their patching. A comparison of these onboard activations with an offline inference of the same model resulted in a 100% agreement, thereby verifying that the competition setup faithfully replicated the software environment and processes on the actual satellite.

Given the tight operational constraints of OPS-SAT, no particular collection strategy for the experimental images could be implemented. Instead, we collected all the images without applying any filter during our allocated time slot and power budget. Consequently, we discarded the results related to those images whose significant distortion made them incomparable to the patches in the OPS-SAT paper and impossible to classify even through visual inspection.

Among the images of sufficient quality, examples that display areas almost completely covered in clouds are often found. The inference of the patches of those cloudy images is primarily correct, assigning the cloud label, with sporadic errors due to confusion with the Water and Snow classes. Fortuitously, a few images were captured showing only partial coverage with clouds, presenting a more diverse set of land-cover features, including mountains, glaciers, water surfaces, meandering rivers, natural vegetation, and agricultural areas.

Given that these images were collected after the labeling campaign of the OPS-SAT dataset, we missed ground-truth labels that would have enabled us to compare the results with respect to the metrics used in the competition. We based our evaluation on opportunistic observations instead, showcasing positive and negative examples for cases in which the correct land-cover type is evident without rigorous expert labeling. Furthermore, as shown herein, the figures related to the in-orbit experiments were enhanced in contrast to aid the reader in discerning the different features. All onboard inference was performed on raw image data. For example example, Fig. 10 shows the image with the highest amount of different classes (according to the Inovor Technologies model).

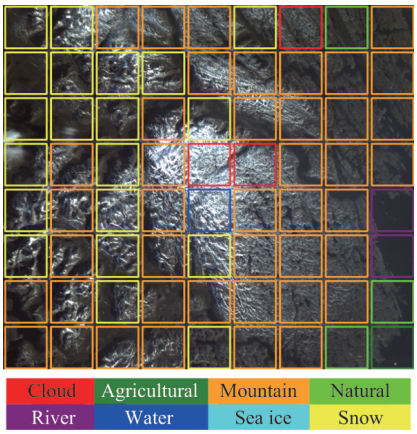


Fig. 10 Example of onboard inference for image patches using the winning model of the OPS-SAT competition.

The model inference reflects the dominating features of mountains and snow, although they are only occasionally accurate. For a few patches, other classes such as Cloud, Water, and Natural are confused with Snow and Mountain. While the cropping of the satellite images for the original experiment did not overlap (including a boundary of 10 px around each patch), a large number of overlapping patches can be constructed from the image, and each of them can be classified independently with the model; this is also possible on board. When merging the results of overlapping patches for a specific label with each other, the classification task can be utilized to perform a coarse image segmentation with some more interpretable results.

Figure 11 shows a comparison between the classification of the original cropping and the proposed coarse segmentation, which allows for the identification of cloud pixels with fair accuracy. Figure 12 shows a reasonable segmentation of an image into the Cloud, Mountain, and Natural classes.

This segmentation technique is capable of separating small-scale features such as potential locations of rivers,

which could otherwise be lost when applying the more rigid non-overlapping cropping. Figure 13 shows examples of (partially) successful and unsuccessful river segmentation.

8 Discussion

8.1 Successful techniques

A comparison of the different solution strategies of the top 3 teams from Section 6.2 shows certain similarities, of which the manual labeling of the unlabeled part of the OPS-SAT dataset is notable. While it is unknown whether lower-ranked teams relied on manual labeling as well, this could be considered an indication that auto-labeling techniques that completely avoid a human in the loop have not been competitive. This hypothesis is corroborated by the performance of our competition's baseline solution, which was obtained by using a variant of FixMatch without any manual labeling, achieving an inferior score of $\mathcal{L}_{\text{quant}} = 0.5397$. Manual labeling

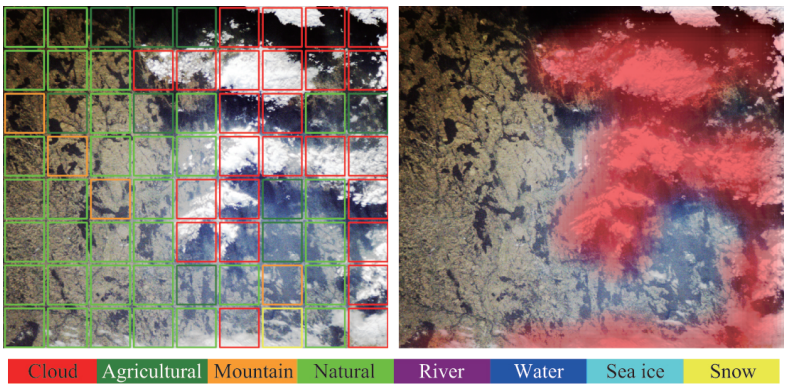


Fig. 11 Left: onboard classification of separated image patches. Right: segmentation of the cloud-class using overlapping patches by sliding a 200 px × 200 px window with a stride of 20 px and adding red color whenever the patch was classified as Cloud.

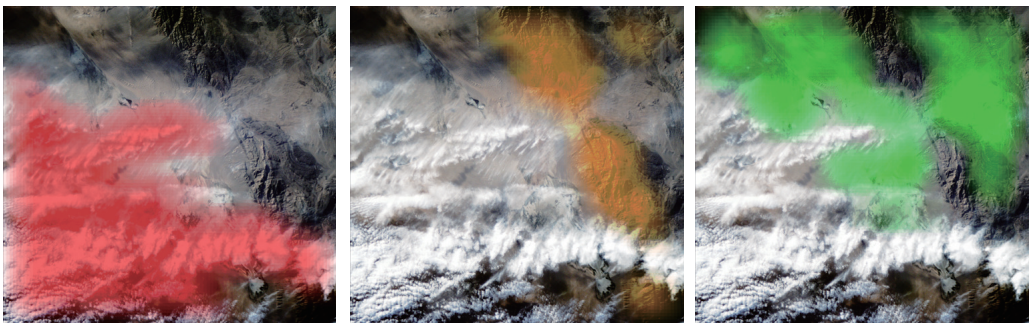


Fig. 12 From left to right: segmentation of clouds (red), mountains (orange), and natural vegetation (green) for onboard model.

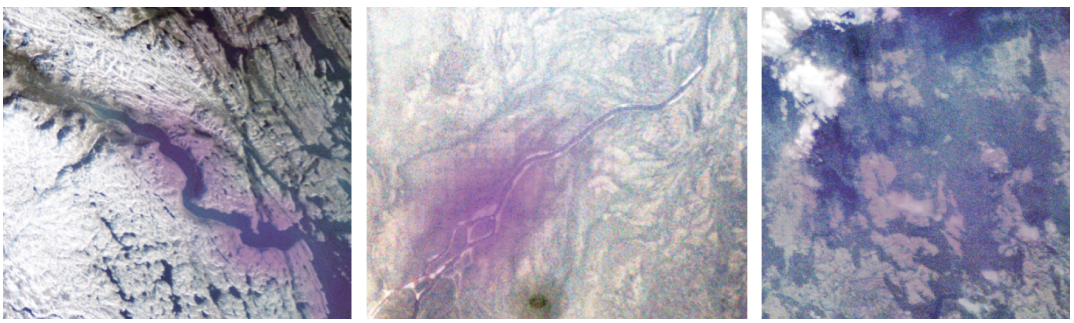


Fig. 13 From left to right: purple region accurately segmenting a broad river; purple region only partially covering the river (uncovered regions have been classified as Natural); false positive: water-surface classified as River.

was not explicitly discouraged for this competition as it would have been practically impossible to prevent it. Nevertheless, we reckon that human annotations, despite their high quality, remain too costly and will not be able to scale with the increasing amount of data collected and the strong demand for ML models in space. Thus, the results of the competition highlight the need to improve the state-of-the-art in semi-supervised learning and fully automated labeling techniques.

One of our goals in setting up the OPS-SAT challenge as a data-centric competition was to explore whether the competitors would be able to exploit openly available external datasets to improve their scores. For the first and second-ranked teams, this appears to be the case. The first-ranked team used data from the official OPS-SAT Flickr webpage, and the 2nd ranked team utilized the NWPU-RESISC45 [39] dataset. While NWPU-RESISC45 data were obtained from a completely different optical sensor, the Flickr images of OPS-SAT were post-processed before publication, making them remarkably different from their unpublished raw data equivalent. Despite these complications, both teams applied different techniques to modify the external data to align with the general properties found in the OPS-SAT dataset. Given their success in the competition, we reckon that making satellite data freely available may be a critical factor in improving machine learning pipelines for onboard space applications in general.

Conversely, the third-ranked team, Ubotica, achieved a highly competitive score by utilizing only the provided dataset. While the other two teams utilized a vast number of different data augmentation techniques, including flips, rotations, contrast enhancements, saturation shifts, and brightness changes, Ubotica limited their approach to flips and rotations. Thus, having a comparatively simple approach in regard to data augmentation, the use of ensembles consisting of additional network architectures different from the EfficientNet-lite0 for pseudo-labeling is a creative approach to circumvent the restrictions on the model architecture that we imposed by the competition design. Given their relative success, this approach could be considered promising, absorbing the benefits of more sophisticated and potentially easier trainable models while ultimately obeying the architecture constraints imposed by the satellite environment. Thus, the impact of high-quality pseudo-labeling and advanced neural network training techniques such as the ensemble implemented by Ubotica are worth considering. Nevertheless, definitive conclusions on the impact of specific techniques would require systematic ablations studies, which are out of the scope of this study.

However, concerning the goal of the data-centric competition as a means to explore whether sparsely labeled data can already deliver meaningful results, we interpret the success of the various approaches over our baseline positively. The use of external datasets, data augmentation techniques, and sophisticated training methods are all viable options when confronted with the development of onboard-AI from sparse data under the limitations of the space environment.

8.2 Class-specific performance

It is worth investigating the most common sources of errors among the top 3 competitors by comparing the results shown by their confusion matrices shown in Fig. 8. Cloud and Sea ice feature a higher recall—(0.877, 0.9389, 0.8421) and (0.8108, 0.8108, 0.9189) for the top 3 respective competitors, respectively. Except for Capgemini, Sea ice also features—together with Water—the highest precision, namely, (0.8824, 0.6818, 0.8947) and (0.8913, 0.9875, 0.7748), respectively. The precision of the Cloud class is generally lowered because of the relatively high number of Snow and Mountain patches predicted as Cloud for all three competitors. The confusion between Snow and Cloud should be not surprising, given the similarity of the features between these two classes. The Mountain class features a maximum precision and recall of 0.5977 and 0.5253 due to the confusion with the Snow and Cloud classes. This fact is partially addressable to the presence of snow or marble in some of the Mountain patches in the test set and the fact that many of the Snow patches are located in mountainous areas. Notably, only Mountain patches free of snow or marble were provided in the train split, which significantly curtailed the representativeness of the training set, introducing a bias error. This is due to the procedure used to select the training patches, which favored the ones showing a complete agreement among the labelers that led to exclude those with confusing features such as snow and marble. Judging from the precision and recall of these classes, this bias was not compensated by any of the techniques of the top 3 competitors. However, all their models tend to over-predict the Cloud class to

the detriment of Mountain and Snow classes, resulting in a high number of false positives.

All three models also tend to predict a significant number of the Agricultural patches as Natural, leading to a low recall for the Agricultural class. For the Capgemini SE model, the recall for Agricultural is even zero.

Lastly, concerning the prediction of the class River, only Inovor Technologies reached a recall higher than 0.45. In most cases, most of the misclassified River patches are images in which the river's width is much smaller compared to the patch size. Therefore, it is reasonable to assume that the co-presence of other elements with more prominent features in those patches led the model to misclassify them.

Figure 14 shows several examples of misclassified patches from the test test under the inference of the model from Inovor Technologies.

8.3 Onboard experiments

When discussing the results of the post-competition orbital experiments, the following aspect must be considered: the OPS-SAT challenge was purposely designed to expose its participants to serious difficulty in order to faithfully simulate the many constraints and hurdles that are expected for the development of onboard AI in space in the near future. Our incentive was to motivate the competitors to find innovative solutions, deploy advanced data augmentation techniques, and make the most of a minimum of data. While releasing the fully labeled dataset would have arguably allowed for better training of the ML model, resulting in potentially higher classification accuracy, it would have compromised our vision to demonstrate that meaningful results can be obtained using remarkably little labeled data. Consequently, the experimental results need to be judged considering this context.

Accounting for the fact that most spacecraft hardware and software differ from their commercial pendants, the powerful computer on OPS-SAT has proven convenient for development and deployment. By supporting widespread and accepted model formats such as TensorFlow Lite [24], the experts that we attracted by organizing our competition were capable of developing a space-ready and functional product without specific knowledge about the OPS-SAT system or direct access to it. The transfer of the winning model to the actual satellite worked flawlessly as our competition setup faithfully replicated the important onboard processes from OPS-SAT. Thus, a widespread adaptation of powerful hardware capable of running widely accepted and established machine learning frameworks should inspire the development of future systems.

Judging the overall quality of the onboard experiments is difficult because of the lack of ground truth and other reasons mentioned in Section 7. The overall performance apparently depends on factors that are difficult to control, as the unprocessed raw sensor data show a high variability. While the competition was specifically designed to work on the raw sensor data, it can be argued that the architecture would be easier to train on more regular data. While high-quality EO products are incrementally refined through pipelines of sophisticated

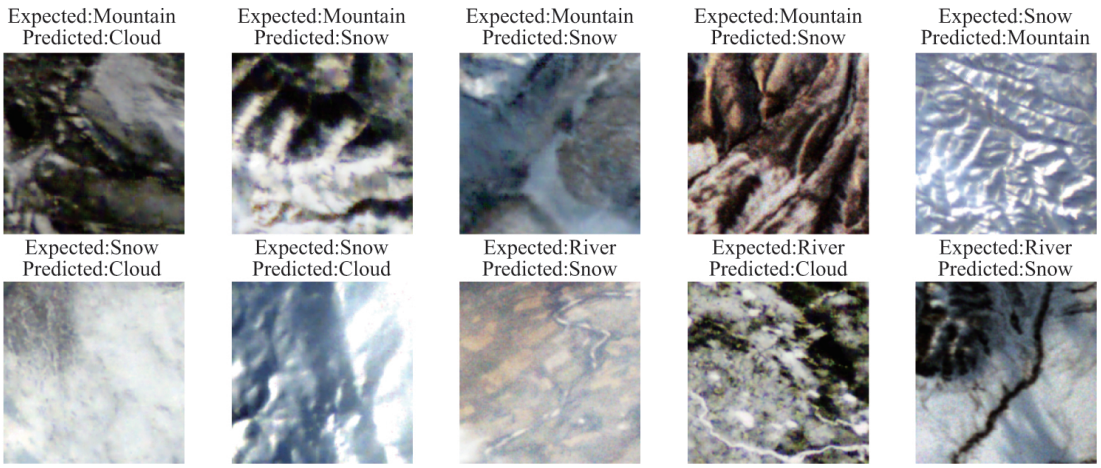


Fig. 14 Misclassified patches by the Inovor Technologies model. Expected vs. predicted labels are shown. A post-processed version of the patches is displayed to facilitate their visualization.

image processing steps that would be cumbersome to execute on board, several basic operations including white-balancing, removal of scattering, and contrast enhancement might be feasible as a pre-processing step on platforms such as OPS-SAT, for example, by using frameworks such as OpenCV [51]. If successful, this could increase the robustness of the model, make it more applicable to difficult conditions, and avoid a few of its shortcomings observed during the orbital experiments.

Nevertheless, when conditions were favorable, the tested model was capable of delivering a decent image segmentation of clouds or other land-cover-types. Surprisingly, the model was also capable of detecting geometric features like rivers in some cases, as long as they were clearly resolved by the sensor. This shows that even small-scale models such as our selection of EfficientNet-lite0 can perform complex tasks in space. Having such and similar capabilities on board a satellite could constitute valuable benefits by enabling increased satellite autonomy. Detecting only the most important and interesting parts of larger images by means of classification and segmentation could allow for optimizing and managing operational constraints including communication and storage demands.

9 Conclusions

We herein present the design and results of "the OPS-SAT case" competition, which was deployed on ESA's Kelvins competition platform to investigate how ML models for in-orbit applications can be trained with access to only minimal raw labeled data. Conducting the training of an actual onboard ML model as a data-centric offline competition allowed us to harness the competitiveness, creativity, and strenuous efforts of various expert teams worldwide. A key enabler for this approach was the reconfigurable computing architecture of OPS-SAT, which offered exceptionally high computational power and the capability to utilize conventional open-source software environments in space. The post-competition analysis of the most successful participants' methodologies revealed a significant tendency to still deploy a human in the loop for labeling uncertain examples; however, a trend towards semi-supervised learning and data augmentation techniques is visible.

With the development of future few-shot learning AI systems, we can expect that onboard deployment will gradually shift to follow the pattern that we propose herein. Even further advances may be anticipated, potentially utilizing the actual satellite as part of an iterative development cycle. The advantages of such a rapid deployment are clear: shortening the data collection period of the satellite and reducing the costs and complications involved with labeling campaigns.

On the basis of our EO application task, we conclude that the emulation of OPS-SAT, including all its operational limitations, was successful. The direct transfer of the trained ML model to the satellite produced several meaningful observations, demonstrating that the best models produced during the competition phase are equally effective in space. Consequently, we encourage the community to follow this example by providing more open data collected in space and allowing for fixed ML models to be readily deployed on future satellites.

Data availability

The training[⑥] and test[⑦] datasets used for "the OPS-SAT case" Kelvins competition are publicly available on Zenodo.

Acknowledgements

The authors would like to thank David Evans, Georges Laebrèche, Sam Bammens, and Vladimir Zelenevskiy for providing data and support in the preparation of "the OPS-SAT case" competition and in the running of the onboard satellite experiments.

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article. The author Dario Izzo is the Editor-in-Chief of this journal.

References

[1] 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.

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

⑥ https://zenodo.org/records/6524750
⑦ https://zenodo.org/records/10301862

on-board deep neural network demonstrator for satellite earth observation. *IEEE Transactions on Geoscience and Remote Sensing*, **2022**, 60: 1–14.

[3] Mateo-Garcia, G., Veitch-Michaelis, J., Purcell, C., Longepe, N., Reid, S., Anlind, A., Bruhn, F., Parr, J., Mathieu, P. P. In-orbit demonstration of a re-trainable machine learning payload for processing optical imagery. *Scientific Reports*, **2023**, 13(1): 10391.

[4] Růžička, V., Vaughan, A., De Martini, D., Fulton, J., Salvatelli, V., Bridges, C., Mateo-Garcia, G., Zantedeschi, V. RaVÆn: Unsupervised change detection of extreme events using ML on-board satellites. *Scientific Reports*, **2022**, 12(1): 16939.

[5] 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.

[6] 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.

[7] Izzo, D., Märtens, M., Pan, B. F. A survey on artificial intelligence trends in spacecraft guidance dynamics and control. *Astrodynamics*, **2019**, 3(4): 287–299.

[8] Galli, A., Giardina, P., Guta, M., Lossi, L., Mancina, A., Moscato, V., Patrone, F., Roseti, C., Romano, S. P., Sperlí, G., *et al.* AI for zero-touch management of satellite networks in B5G and 6G infrastructures. In: Proceedings of the International Workshop on Artificial Intelligence in beyond 5G and 6G Wireless Networks, **2022**.

[9] Ferreira, P. V. R., Paffenroth, R., Wyglinski, A. M., Hackett, T. M., Bilen, S. G., Reinhart, R. C., Mortensen, D. J. Reinforcement learning for satellite communications: From LEO to deep space operations. *IEEE Communications Magazine*, **2019**, 57(5): 70–75.

[10] Derksen, D., Meoni, G., Lecuyer, G., Mergy, A., Märtens, M., Izzo, D. Few-shot image classification challenge on-board OPS-SAT. In: Proceedings of the 35th Conference on Neural Information Processing Systems, **2021**.

[11] Sebastianelli, A., Del Rosso, M. P., Ullo, S. L. Automatic dataset builder for Machine Learning applications to satellite imagery. *SoftwareX*, **2021**, 15: 100739.

[12] Kikaki, K., Kakogeorgiou, I., Mikeli, P., Raitsos, D. E., Karantzalos, K. MARIDA: A benchmark for Marine Debris detection from Sentinel-2 remote sensing data. *PLoS One*, **2022**, 17(1): e0262247.

[13] 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.

[14] Richards, J. A. *Remote Sensing Digital Image Analysis*. Springer Cham, **2022**.

[15] Izzo, D., Meoni, G., Gómez, P., Dold, D., Zoechbauer, A. Selected trends in artificial intelligence for space applications. *arXiv preprint*, **2022**, arXiv:2212.06662.

[16] Kisantal, M., Sharma, S., Park, T. H., Izzo, D., Martens, M., D'Amico, S. Satellite pose estimation challenge: Dataset, competition design, and results. *IEEE Transactions on Aerospace and Electronic Systems*, **2020**, 56(5): 4083–4098.

[17] Park, T. H., Martens, M., Lecuyer, G., Izzo, D., D'Amico, S. SPEED+: Next-generation dataset for spacecraft pose estimation across domain gap. In: Proceedings of the IEEE Aerospace Conference, **2022**: 1–15.

[18] Märtens, M., Izzo, D., Krzic, A., Cox, D. Super-resolution of PROBA-V images using convolutional neural networks. *Astrodynamics*, **2019**, 3(4): 387–402.

[19] Chen, B., Liu, D. Q., Chin, T. J., Rutten, M., Derksenv, D., Martens, M., von Looz, M., Lecuyer, G., Izzo, D. Spot the GEO satellites: From dataset to Kelvins SpotGEO challenge. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, **2021**: 2086–2094.

[20] Evans, D., Labrèche, G., Mladenov, T., Marszk, D., Zelenevskiy, V., Shiradhonkar, V. OPS-SAT LEOP and commissioning: Running a nanosatellite project in a space agency context. In: Proceedings of the 36th Annual Small Satellite Conference, **2022**: SSC22-IX-06.

[21] Abderrahmane, N., Miramond, B., Kervennic, E., Girard, A. SPLEAT: SPiking Low-power Event-based ArchiTecture for in-orbit processing of satellite imagery. In: Proceedings of the International Joint Conference on Neural Networks, **2022**: 1–10.

[22] Labrèche, G., Evans, D., Marszk, D., Mladenov, T., Shiradhonkar, V., Zelenevskiy, V. Artificial intelligence for autonomous planning and scheduling of image acquisition with the SmartCam app on-board the OPS-SAT spacecraft. In: Proceedings of the AIAA SCITECH Forum, **2022**: AIAA 2022-2508.

[23] Labrèche, G., Evans, D., Marszk, D., Mladenov, T., Shiradhonkar, V., Soto, T., Zelenevskiy, V. OPS-SAT spacecraft autonomy with TensorFlow lite, unsupervised learning, and online machine learning. In: Proceedings of the IEEE Aerospace Conference, **2022**: 1–17.

[24] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., *et al.* TensorFlow: Large-scale machine learning on heterogeneous systems. **2015**. Software available from tensorflow.org

[25] Park, T. H., Märtens, M., Jawaid, M., Wang, Z., Chen,

B., Chin, T. J., Izzo, D., D'Amico, S. Satellite pose estimation competition 2021: Results and analyses. *Acta Astronautica*, **2023**, 204: 640–665.

[26] Uriot, T., Izzo, D., Simões, L. F., Abay, R., Einecke, N., Rebhan, S., Martinez-Heras, J., Letizia, F., Siminski, J., Merz, K. Spacecraft collision avoidance challenge: Design and results of a machine learning competition. *Astrodynamics*, **2022**, 6(2): 121–140.

[27] Derksen, D., Meoni, G., Lecuyer, G., Mergy, A., Märtens, M., Izzo, D. The OPS-SAT case dataset (1.0.0). Zenodo, **2022**, https://doi.org/10.5281/zenodo.6524750

[28] Derksen, D., Meoni, G., Lecuyer, G., Mergy, A., Märtens, M., Izzo, D. The OPS-SAT case: Test dataset (1.0.0). Zenodo, **2023**, https://doi.org/10.5281/zenodo.10301862

[29] Grizonnet, M., Michel, J., Poughon, V., Inglada, J., Savinaud, M., Cresson, R. Orfeo ToolBox: Open source processing of remote sensing images. *Open Geospatial Data, Software and Standards*, **2017**, 2(1): 15.

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

[31] Liu, R. Higher accuracy on vision models with EfficientNet-Lite. **2020**. Available at https://blog.tensorflow.org/2020/03/higher-accuracy-on-vision-models-with-efficientnet-lite.html

[32] Musial, J. P., Bojanowski, J. S. Comparison of the novel probabilistic self-optimizing vectorized earth observation retrieval classifier with common machine learning algorithms. *Remote Sensing*, **2022**, 14(2): 378.

[33] Deshpande, P., Belwalkar, A., Dikshit, O., Tripathi, S. Historical land cover classification from CORONA imagery using convolutional neural networks and geometric moments. *International Journal of Remote Sensing*, **2021**, 42(13): 5144–5171.

[34] Navarro, A., Silva, I., Catalão, J., Falcão, J. An operational Sentinel-2 based monitoring system for the management and control of direct aids to the farmers in the context of the Common Agricultural Policy (CAP): A case study in mainland Portugal. *International Journal of Applied Earth Observation and Geoinformation*, **2021**, 103: 102469.

[35] Sohn, K., Berthelot, D., Li, C. L., Zhang, Z. Z., Carlini, N., Cubuk, E. D., Kurakin, A., Zhang, H., Raffel, C. FixMatch: Simplifying semi-supervised learning with consistency and confidence. In: Proceedings of the 34th International Conference on Neural Information Processing Systems, **2020**: 596–608.

[36] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., *et al.* PyTorch: An imperative style, high-performance deep learning library. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems, **2019**: 8026–8037

[37] European Space Agency. the_opssat_case_starter_kit. **2022**. Available at https://gitlab.com/EuropeanSpaceAgency/the_opssat_case_starter_kit

[38] Keras. **2015**. Available at https://keras.io

[39] Cheng, G., Han, J. W., Lu, X. Q. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, **2017**, 105(10): 1865–1883.

[40] Chollet, F. Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, **2017**: 1800–1807.

[41] Pitié, F., Kokaram, A. C., Dahyot, R. Automated colour grading using colour distribution transfer. *Computer Vision and Image Understanding*, **2007**, 107(1–2): 123–137.

[42] Papadakis, N., Provenzi, E., Caselles, V. A variational model for histogram transfer of color images. *IEEE Transactions on Image Processing*, **2011**, 20(6): 1682–1695.

[43] Rabin, J., Ferradans, S., Papadakis, N. Adaptive color transfer with relaxed optimal transport. In: Proceedings of the IEEE International Conference on Image Processing, **2014**: 4852–4856.

[44] Ferradans, S., Papadakis, N., Peyré, G., Aujol, J. F. Regularized discrete optimal transport. *SIAM Journal on Imaging Sciences*, **2014**, 7(3): 1853–1882.

[45] Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., Kalinin, A. A. Albumentations: Fast and flexible image augmentations. *Information*, **2020**, 11(2): 125.

[46] Lin, T. Y., Goyal, P., Girshick, R., He, K. M., Dollar, P. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **2020**, 42(2): 318–327.

[47] Yeung, M., Sala, E., Schönlieb, C. B., Rundo, L. Unified focal loss: Generalising Dice and cross entropy-based losses to handle class imbalanced medical image segmentation. *Computerized Medical Imaging and Graphics*, **2022**, 95: 102026.

[48] Müller, R., Kornblith, S., Hinton, G. When does label smoothing help? In: Proceedings of the 33rd International Conference on Neural Information Processing Systems, **2019**: 4694–4703.

[49] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, **2016**: 2818–2826.

[50] Barz, B., Denzler, J. Deep learning on small datasets without pre-training using cosine loss. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision, **2020**: 1360–1369.

[51] Bradski, G. The openCV library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, **2000**, 25(11): 120–123.

Gabriele Meoni, Ph.D., is an assistant professor in the Faculty of Aerospace Engineering of Delft University of Technology. From September 2021 until April 2023 he was an internal research fellow the Φ-lab division of the European Space Agency (ESA). During October 2022 until March 2023 he was visiting researcher in AI Sweden. He was former research fellow of the ESA Advanced Concepts Team, which he joined in the 2020 after receiving his Ph.D. degree from University of Pisa (supervisor Prof. Luca Fanucci) in information engineering. His research topics of interest include satellite onboard processing, embedded computing systems, and edge computing.

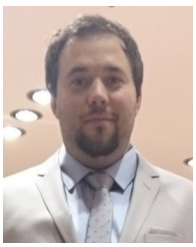
Marcus Märtens has been working as scientific crowd-sourcing engineer and internal research fellow at the Advanced Concepts Team of the European Space Agency (ESA) from February 2018 to 2022. He holds an HUMIES gold medal for developing algorithms achieving human competitive results in trajectory optimization. He designed and conducted multiple open scientific competitions centered around artificial intelligence application for space, including satellite super-resolution and pose estimation at ESA. In 2018, he received his Ph.D. degree from the Delft University of Technology for his work on information propagation in complex networks. Marcus has worked together with experts from different fields and authored works related to space, neuroscience, cyber-security and gaming.

Dawa Derksen received his master degree in aerospace engineering from the Institut Supérieur de l'Aéronautique et de l'Espace (ISAE-Supaéro), Toulouse, France, in 2016, and his Ph.D. degree from the the Centre d'Etudes Spatiales de la Biosphère (CESBIO) Laboratory, Toulouse, France, in 2019. His Ph.D. topic was the operational production of image processing algorithms applied to the large scale classification of the Earth observation images for land cover mapping.

Kenneth See received his bachelor degree with honours in aerospace engineering and his bachelor degree in computer science in 2020 from the University of Adelaide. From 2020 to 2022, he worked as a modelling and simulation engineer at Inovor Technologies. Currently, Kenneth is a research engineer at Lockheed Martin Australia, STELaRLab. His research interests include orbit determination, state estimation, tracking, and fusion.

Toby Lightheart received his B.Eng. degree from the University of Tasmania, Australia, in 2008. He completed his Ph.D. degree at the University of Adelaide, Australia, in 2018 on constructive algorithms for artificial neural networks and approximations of neuroplasticity. He worked at Inovor Technologies on nanosatellites and simulation and modelling. He currently works at the Australian Government Department of Defence.

Anthony Sécher received his Ph.D. degree in prehistory and archaeological sciences from the University of Bordeaux in 2017. He, then, joined in 2021 Capgemini Engineering's R&I Department in Blagnac, in the Hybrid Intelligence team. His work is part of the ND2I research project and focuses on new applications of computer vision to soil recognition.



Arnaud Martin received his Ph.D. degree in artificial intelligence. Since 2012, he is a senior data scientist and was Tech Lead IA/Deep Learning France in the team Hybrid Intelligence of Capgemini Engineering. He is the leader for the whole of France in the field of AI and more specifically deep learning, design/adaptation of intelligent systems, using and creating new techniques from the field of artificial intelligence, mainly deep learning, on both GPU and edge servers.

David Rijlaarsdam has his Master of Science degree in aerospace engineering from the Delft University of Technology with a specialization in space system engineering. He currently is a senior space system engineer for Ubotica Technologies, where he manages the space system research group. He has previously been

part of the automation and robotics section of the European Space Agency and part of the advanced architecture team of Intel Movidius.

Vincenzo Fanizza graduated with his master degree in aerospace engineering at Delft University of Technology. He worked as an intern at Ubotica Technologies, where he learned to develop systems based on artificial intelligence and apply machine learning to space imagery. His interests are related to the general application of AI and ML to space missions, ranging from the Earth observation to spacecraft relative navigation.

Dario Izzo received his doctoral degree in aeronautical engineering from the University Sapienza of Rome, Rome, Italy, in 1999, his second master degree in satellite platforms from the University of Cranfield, Bedford, UK, in 2002, and his Ph.D. degree in mathematical modeling from the University Sapienza of Rome, in 2003. He lectured classical mechanics and space flight mechanics with the University Sapienza of Rome. He then joined the European Space Agency, Noordwijk, the Netherlands, where he later became the scientific coordinator with the Advanced Concepts Team. He devised and managed the Global Trajectory Optimization Competitions events, the ESA Summer of Code in Space, and the Kelvins innovation and competition platform. He authored or coauthored more than 170 papers in international journals and conferences making key contributions to the understanding of flight mechanics and spacecraft control and pioneering techniques based on evolutionary and machine learning approaches. Dr. Izzo received the Humies Gold Medal and led the team winning the eighth edition of the Global Trajectory Optimization Competition.