SpaceOps-2025, ID # 307

# Advancing AI Reliability for Space Missions through Machine Learning Operations

**Becca-Bonham-Carter[a], Alexis Pascual[b], Tim Heydrich[c], Luis Chavier[d], Ian Abcarius[e], Sophie Villemure[f], Galen O'Shea[g], Hugo Burd[h], Shannon Paul[i], Alan Higginson[j], Matt Cross[k], Samara Pillay[l], Yolanda Brown[m], Michele Faragalli[n], Andrew J. Macdonald[o*]**

[a] *Mission Control, Canada, becca@missioncontrolspace.com*
[b] *Mission Control, Canada, alexis@missioncontrolspace.com*
[c] *Mission Control, Canada, tim@missioncontrolspace.com*
[d] *Mission Control, Canada, luis@missioncontrolspace.com*
[e] *Mission Control, Canada, ian@missioncontrolspace.com*
[f] *Mission Control, Canada, sophie@missioncontrolspace.com*
[g] *Mission Control, Canada, galen@missioncontrolspace.com*
[h] *Mission Control, Canada, hugo@missioncontrolspace.com*
[i] *Mission Control, Canada, shannon.paul@missioncontrolspace.com*
[j] *Mission Control, Canada, alan@missioncontrolspace.com*
[k] *Mission Control, Canada, cross@missioncontrolspace.com*
[l] *Mission Control, Canada, samara@missioncontrolspace.com*
[m] *Mission Control, Canada, yolanda@missioncontrolspace.com*
[n] *Mission Control, Canada, michele@missioncontrolspace.com*
[o*] *Mission Control, Canada, macdonald@missioncontrolspace.com*
* Corresponding Author

**Abstract**

Core to the goal of increasing perception and autonomy capabilities in space missions is the development of techniques and standards to characterize the robustness and reliability of Artificial Intelligence (AI) sub-systems in flight and ground segments. Methods developed in the terrestrial sector, such as Machine Learning Operations (MLOps), incorporate the principles of Continuous Integration, Continuous Deployment, and Continuous Training to counter time-dependent errors such as domain shifts that emerge in operational AI systems. Mission Control, with the support of the Canadian Space Agency, is developing and launching Mission Persistence in 2025 to introduce and test new approaches to AI in space operations. Mission Control's technology demonstration will maintain onboard AI models in-orbit through the use of an MLOps pipeline in the ground segment that divides pre-production data and model elements from a production software environment that monitors model performance on newly acquired earth observation data and re-trains and deploys updated models. This system will be tested and characterized over the course of one year of operations and compared to the performance of static models that are left in their initial, pre-launch state. Through this in-orbit testbed Mission Control will pioneer new AI approaches to space operations for earth observation, rendezvous and proximity operations, and lunar surface missions.

**Keywords:** Artificial Intelligence (AI), Machine Learning Operations (MLOps), Earth Observation (EO), New Observing Strategies (NOS), onboard processing, domain adaptation

## Nomenclature

*D = Domain of a generic feature and label space*
*$P_D(X)$= Marginal probability distribution of the Feature Space X in Domain D*
*$P_D(X|Y)$ = Conditional probability distribution of the Feature Space X given the Label Space Y in Domain D*
*$REFLECTANCE_{TOA}$ = Reflectance measured at top of atmosphere*
*X = Feature Space*
*Y = Label Space*

## Acronyms/Abbreviations

AI = Artificial Intelligence
API = Application Programming Interface
CSA = Canadian Space Agency
EO = Earth Observation
ESA = European Space Agency

GradCAM = Gradient-weighted class activation mapping
GSN = Ground Station Network
HITL = Hardware in the Loop
LCCS = Land-Cover Classification System
MLOps = Machine Learning Operations
MODIS = Moderate Resolution Imaging Spectroradiometer
NOS = New Observing Strategies
ORT = Onnxruntime
TRL = Technology Readiness Level
V&V = Verification and Validation

## 1. Introduction

The use of deep learning algorithms onboard space systems presents opportunities for new autonomous applications in both earth orbiting and deep space missions. These new opportunities, using onboard processing to circumvent limitations in bandwidth by processing data at the point of acquisition, are enabled by increased embedded computing power and the advent of new AI approaches centred around deep neural networks [1]. These advancements come with challenges related to robustness, integrity, and trust in an operational setting. The incorporation of neural networks and deep learning into space flight systems introduces new considerations around technical debt, infrastructure, and data and software validation and verification. These additional technical considerations, which can often be hidden [2], require new solutions to technical problems inherent in data-driven algorithms and new categories to classify the technology, such as the Machine Learning Technology Readiness Levels proposed by Lavin *et al.* [3] or the Space Trusted Autonomy Readiness Levels proposed by Hobbs *et al.* [4] to build robust and trustworthy solutions that can be used in operations. In this paper we describe how a Machine Learning Operations (MLOps) framework, which combines the core principles of continuous integration, deployment, and training, can stabilize performance variability in deep learning models deployed in the flight segment and describe its implementation in an upcoming earth observation satellite mission, Mission Persistence.

In the introduction we provide a background review of the relevant literature on the advancement of AI-enabled satellites capable of providing New Observing Strategies (NOS) and the technical debt and domain adaptation challenges inherent in combing deep learning and satellite operations. In the Material and Methods, we describe the flight system design and operating procedure as well as an MLOps pipeline that can perform updates to machine learning payloads in space. In the Theory section we provide a mathematical description of a specific type of possible performance degradation, covariance shift, which in-orbit machine learning solutions are liable to experience. In the Results and Discussion, we show neural network model results on pre-launch data and discuss how the MLOps pipeline ensures continual training and redeployment of models to orbit, increasing overall robustness and reliability. In the Conclusion we describe the next steps for commissioning and operations and implications for missions beyond low earth orbit.

### *1.1 AI-Enabled Satellites*

Chien *et al.* proposed the concept of a Sensor Web [5] – a distributed system of sensing nodes that are interconnected and capable of a more dynamic and real-time response than traditional Earth Observation satellite missions or constellations. The Sensor Web concept has evolved over the last twenty years as compute and sensing technology has advanced and it is complementary to the concept of New Observing Strategies (NOS) [6], [7], new methods and operations techniques that expand the art-of-the-possible for space-based EO systems. A subset of NOS, and an enabler for future Sensor Web concepts, are satellites that can host machine learning payloads within their flight software, which have shown significant promise in application areas like disaster response, such as flooding [8] and wildfires [9].

These kinds of AI-enabled satellites, capable of running deep neural network inference within machine learning payloads, have been advanced through a combination of government agency and industry owned assets. The European Space Agency (ESA) has been on the leading edge of this latest generation of neural network enabled satellites including ϕ-Sat 1 [10], OPS-SAT [11], and most recently ϕ-Sat 2 [12]. The Australian Space Agency recently launched the Kanyini mission, which includes onboard neural networks trained for fire and smoke detection [13]. In the private sector examples include missions by KP Labs with Intuition-1 [14], OroraTech with FOREST-2 [9], and Ubotica and OpenCosmos with CogniSat-6 [15]. Rijlaarsdam *et al.* [15] provide a thorough chronology of autonomy and AI-enabled earth observation satellite missions, including differences in deployed neural network applications, processing hardware, sensors, and communications links.

### *1.2 Machine Learning Systems and Technical Debt*

The introduction of deep learning technologies can create state-of-the-art solutions to problems in computer vision and pattern recognition. These achievements come at the cost of increases in the extent and depth of technical debt across software, data, serving infrastructure, and configuration management, much of which is beyond the scope of the machine learning model itself, and which can remain hidden until a point of failure emerges [2]. These challenges have led authors to analyse the traditional Technology Readiness Levels (TRLs) commonly used in space technology development and to propose new categories that reflect the additional considerations required for machine learning technology [3]. To sustain continuous and performant deep learning applications onboard a space mission requires flight and ground segment design that encompasses the data, infrastructure, and code management needed to keep ML solutions performant in operations.

Within the data aspect, space missions often do not allow operators to capture fully representative training datasets for deep learning models until commissioning and operations, leaving the use of simulated or synthetic data as the only viable option for model training during pre-launch development and Verification and Validation (V&V) activities. The jump from pre-launch data to operational data is likely to encounter known problems in domain adaptation for machine learning models, such as changes in noise or atmospheric conditions as studied by Nalepa *et al.* [14]. These effects are not unique to deep learning in space-based EO and are observed in fields such as medicine, for example when model outputs vary based on changes in the underlying statistics of the patient distribution, or changes in sensor type, modality, or calibration [16]. In the field of deep learning inference for space-based earth observation and space missions there are open questions about the impact of these domain adaptation challenges. We propose MLOps as a solution, which enables the automated re-training, re-deployment, and continuous monitoring of deep learning algorithmic performance onboard spacecraft during operations. We will test and validate this approach as part of Mission Persistence, a 6U CubeSat being built by Spire Global, which will host Mission Control machine learning payloads and be supported during operations by an MLOps pipeline in the mission ground segment.

## 2. Material and methods

### *2.1 Space System Design*

Figure 1 shows the flight payload design and ground flight interface for Persistence. Spire provides ground and flight interfaces to task the payloads onboard the satellite via a tasking application programming interface (API). The API provides interfaces for determining satellite availability windows and supports tasking of different executables onboard the satellite. The Payload Executable (payload_exec), which runs on a Xilinx Ultrascale+ System-on-a-Chip, is responsible for the initial setup and execution steps when tasking windows are received. It handles the transfer and extraction of data that was uploaded to the satellite as well as calling the subsequent executables specified for the window. Machine learning model weights within the payload are designed to be updated during flight.

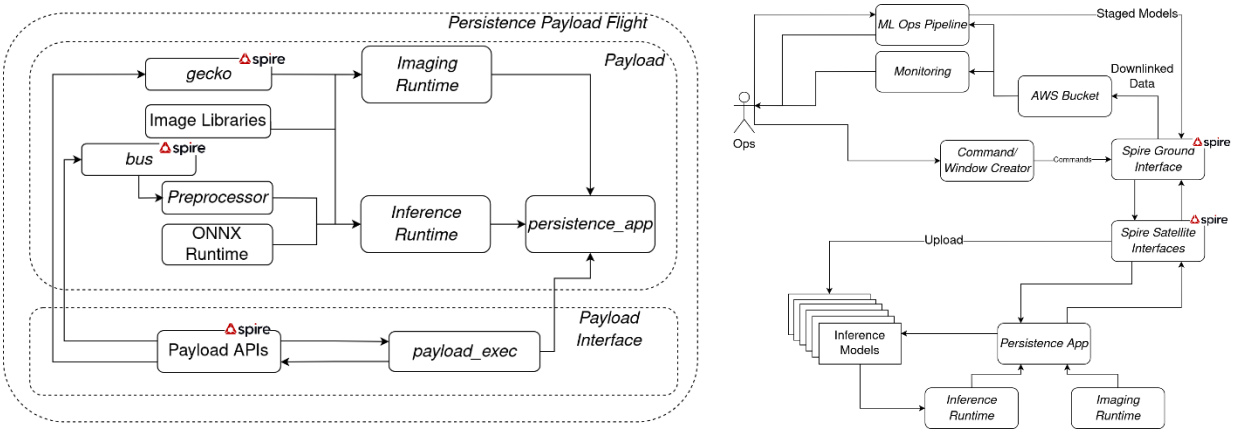


Fig. 1. Persistence flight payload design (left) and Persistence ground-flight interface (right).

The primary EO imaging payload is a Dragonfly Gecko*. The Persistence application (payload_app) handles imaging requests as well as the image preprocessing and inference using user specified neural network models. The payload supports both segmentation and classification models and the use of *onnxruntime* (ORT) enables support of a

* Gecko Imager - Dragonfly Aerospace

variety of model architectures. The payload also supports models that have been optimized through pruning or quantization. The payload is called using a payload config which specifies the imaging parameters as well as which model metadata file to use for inference and any other preprocessing parameters. The imaging side of the payload app calls the Gecko imager. The inference side uses the model details specified in a metadata file to setup the ORT inference engine and load neural network models. It also applies any preprocessing such as tiling or normalizing to the images before running model inference. The payload_app can be updated post launch with the update process described in section 2.1.2.

*2.1.2 Planned Operations Cycle*

Figure 2 depicts the four distinct phases of Mission Control operations of the satellite. During uplink, a tasking window is created on the ground and transmitted to the satellite using the Spire Ground Station Network (GSN). The window configuration file dictates the payload tasking for the next operating period. The first payload task is an imaging session which collects images of Earth's surface and stores them for downlink and data processing. Onboard processing takes place following an imaging session and is broken down into separate image preprocessing and neural network model inference steps. The model outputs provide the satellite with metadata which could be used for satellite optimization and operations. The images and metadata are transmitted to the Spire GSN and into the MLOps pipeline in the ground segment for storage, analysis, and model retraining.

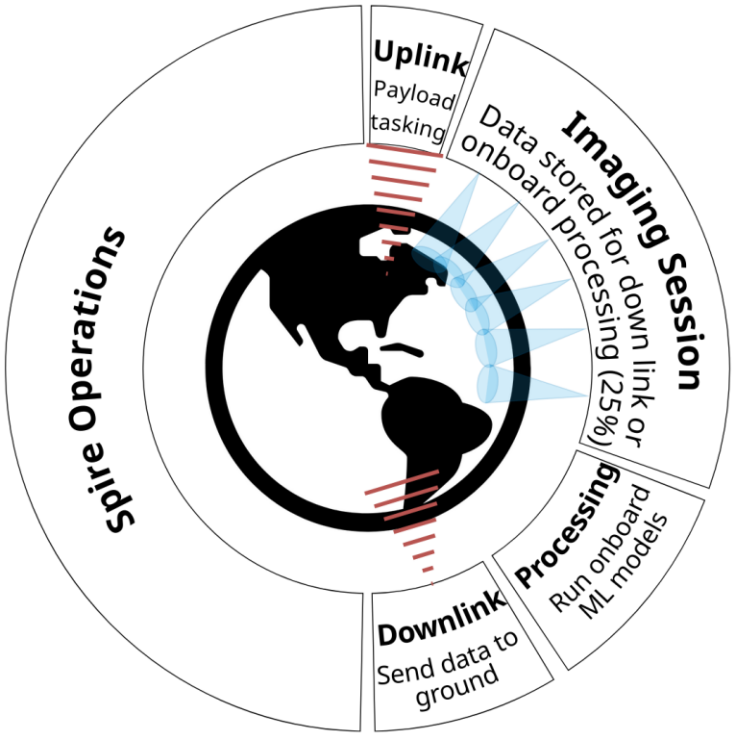


Fig. 2. Mission Persistence operations cycle.

A key component of MLOps is updating the model weights in flight on a regular basis in response to changes or degradation in performance. To optimize the use of limited uplink bandwidth, patch files are used to reduce the upload size required by only updating the parts of the model weights that changed between the different iterations. The steps given below can be used for updating models, metadata files or any other component of the payload application.

1. Create patch files for the files to patch, on the ground.
2. Post upload of patch files.
3. Create window to apply the patches.

*2.2 MLOps Pipeline*

Figure 3 showcases the design of the MLOps pipeline embedded in the Persistence ground segment. This design is an evolution of Mission Control's previous system [17] used to fly neural network models on OPS-SAT [18] and to lunar orbit [19]. This design is broken down into three main segments: Data, Models, and Production. Each segment

is designed to facilitate the rapid and reproducible processing of data and supports the capability to provide neural network updates to maintain model performance across the mission lifecycle.

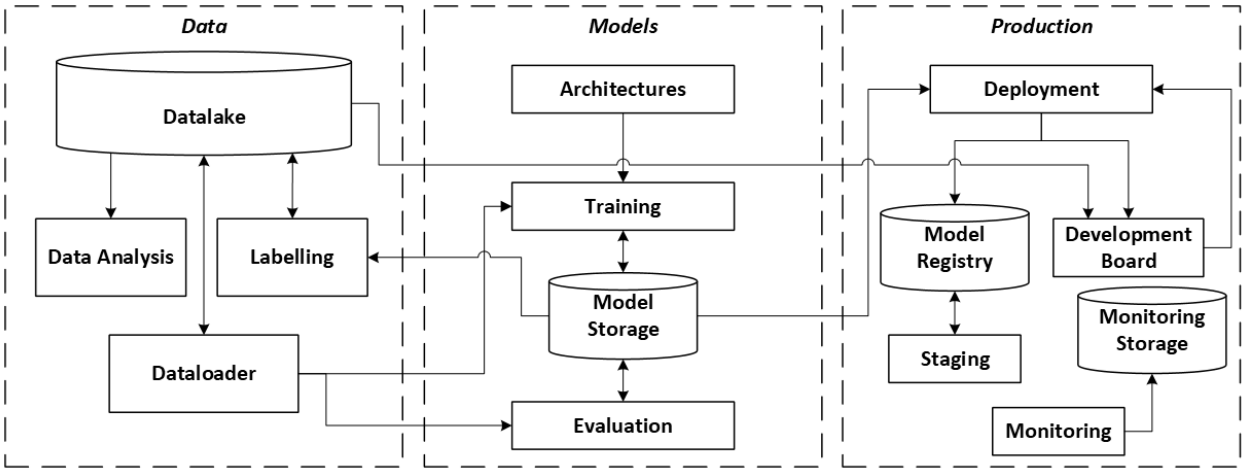


Fig. 3. High level design of the MLOps pipeline.

The Data segment ingests new datasets and inbound data streams from relevant missions. Datasets are stored either as raw, cleaned, or labelled in an S3 bucket. The Data segment includes data analysis tools, preprocessing, and labelling tools.

The Models segment handles training, model evaluation, and architecture definitions. Storing all model-related tools in one place allows operators to quickly produce new and tuned models. This segment's primary output is trained models with traceable training logs, validation metrics, relevant metadata, and trained model checkpoints at each epoch. Third party tools are used for training monitoring and provide insight into training metrics such as loss, accuracy, learning rate, and intersection over union for training and validation data.

The Production segment interfaces with prepared models and metadata for payloads. Models at this stage are stored in a model registry bucket along with metadata files tailored to each mission payload. Production verifies model performance on representative hardware in the form of a development board and characterizes model decay using a model monitoring S3 bucket and interface.

*2.2.2 Dataflow*

The MLOps design provides flexibility in the operation order to accommodate for use cases such as data analysis, model experimentation and data repurposing. Despite this, the primary dataflow path for training a flight model from raw data is straightforward.

1. Clean the raw data and transfer cleaned images to the cleaned section of the Data Lake.
2. Assign ground truth labels for supervised learning, using either human labels or model generated pseudo labels.
3. Human review of the ground truth and transfer of approved ground truth labels to the appropriate section of the Data Lake.
4. Split the labelled dataset into test, validation, and training and generate manifests using the data loader.
5. Train and tune new models for flight deployment.
6. Compile the best performing model to be compatible with the flight payload inference tools.
7. Complete validation and verification testing (V&V) using hardware in the loop testing.
8. Stage the new model with the required model metadata files and add it to the Model Registry so it is accessible by the flight deployment team.

*2.2.3 Deployment and Verification and Validation*

To ensure that quality model products are provided to flight and ground systems, a detailed V&V process is followed before adding a model to the Model Registry. The V&V workflow consists of a series of tests that ensure the models produced by the MLOps pipeline will perform over the mission operation period.

1. Dataset Validation:

   a. Dataset Similarity: Check that the training dataset covers a similar spatial resolution, pixel resolution, product processing level, satellite altitude, and satellite orientation as the Persistence Satellite.
   b. Class Representation: Confirm that the training set contains all Persistence target classes.
2. Model Verification and Validation:
   a. Model Evaluation: Using traditional ground truth evaluation, the model must achieve a designated accuracy. For Persistence this is 70%.
   b. Domain Shift Resistance: Evaluate the model on a small dataset from another satellite with a Gecko Imager to characterize the model's response to the domain shift.
   c. Long-term Simulation: By splitting the dataset into blocks, we can represent periods of additional data being gathered over the course of the mission. By training a model on the first n blocks and evaluating it on the n+1 block, we can simulate the retraining process as new data enters the training set. The model must plateau above 70% accuracy for Persistence models.
   d. Hardware In the Loop (HITL): Using a representative hardware environment and the newest version of the flight software payload, the model must be able to run inference on a testing dataset.

## 3. Theory

### *3.1 Domain Shifts*

Domain shifts that can affect the performance of a machine learning algorithm come in several varieties, including prior shift, concept shift, and covariate shift [16]. The primary domain shift addressed by the design of the MLOps pipeline is covariate shift, which can be described as follows. We consider a domain *D* with a Feature Space *X* and Label Space *Y*. Onboard processing algorithms seek to successfully map elements of the Feature Space to the Label Space $X \rightarrow Y$. Pre-launch data used for training models before flight is drawn from a domain *P* while the target domain during operations is *O*. Covariate shift occurs when the marginal probability distribution of the feature space between the pre-launch and operations environment is not equivalent but the conditional probabilities of the label space between pre-launch and operations is, as captured in Equation 1 and 2:

$$P_P(X) \neq P_O(X) \quad (1)$$

but

$$P_P(Y|X) = P_O(Y|X) \quad (2).$$

### *3.1 Domain Adaptation and MLOps*

Causes of covariate shift include switching imaging modalities, changing sensors within the same sensor modality, or evolution or changes in the population under study. The purpose of domain adaptation is to compensate for discrepancies introduced by any or all of these causes. MLOps enables model weights to respond and learn about the evolution in the distribution of earth observation data taken over the mission, starting with rectifying discrepancies between the feature space in the pre-launch and operations domains but then extending as the operations domain evolves throughout the mission lifetime. A MLOps pipeline helps achieve domain adaptation by continually correcting for time dependent differences in the marginal probability distributions between model training and testing environments.

## 4. Results and Discussion

### *4.1 Pre-launch Training Data*

Prior to launch, representative earth observation data from ESA's Copernicus program was used to characterize the behaviour of the MLOps pipeline and train machine learning models for deployment in the payload software at launch. ESA's Sentinel-2 dataset is one of the most comprehensive, public EO datasets covering all parts of the world [20]. The Sentinel-2 data from SEN12MS [20], [21] is the primary dataset used for the pre-launch training of Persistence models. SEN12MS is a dataset that combines multiple EO datasets for land-cover classification over four meteorological seasons and several classes. We extracted only the Sentinel-2 and Moderate Resolution Imaging Spectroradiometer (MODIS) derived Land-Cover Classification System (LCCS) data as it is the most representative of our satellite environment and use case. MODIS derived LCCS is a comprehensive dataset that contains land-cover classifications of the Sentinel-2 images and provided approximate ground truth for image classification tasks. These images were specially curated to avoid cloud cover images, however, since Persistence models will be run in flight, they need to classify images with high cloud cover. To fill this gap in the training data, we collected an additional 9714 tiles of size 256x256 that were hand selected and labelled from the Copernicus browser [22] to make up the cloud class. Lastly, a snow class was curated from living atlas's Sentinel-2 explorer [23] consisting of 2953, 256x256 pixel

tiles. To closely resemble the data received from the Gecko imager, all training data is derived from Sentinel-2 L1C products. To convert the spectral imagery bands to true colour images, we started by applying Equation 3 to calculate top-of-atmosphere reflectance:

$$\frac{DN}{10000} = REFLECTANCE_{TOA} \quad (3).$$

From there to convert the data into 8-bit unsigned integers representing RGB values we applied Equation 4:

$$TrueColor_{R|G|B} = clamp(A \cdot REFLECTANCE_{TOA} \cdot 255) \quad (4).$$

Here, $A$ is a gain that was found to be roughly 2.5 to make the RGB values in our L1C True Colour Images. These equations were derived from the expected range of Sentinel-2 optical band values [24]. After manually filtering out any solid black or white images, we were left with a pre-launch dataset of 189,159 256x256 RGB True Colour Images.

Clustering methods were used to attempt class definitions, however the pretrained models failed to sufficiently separate classes. This led us to use MODIS LCCS data to define our labelling taxonomy. Eleven class labels were defined for an image classification task wherein each of the 256x256 pixel image tiles would be assigned a class label by a representative onboard neural network. A comparison of different architectures for both floating point and fixed-point quantized models led to the choice of a MobileNetv3 backbone [25], with overall accuracy on the test set of 86.9% and strong performance across all classes, as captured in the normalized confusion matrix in Figure 4.

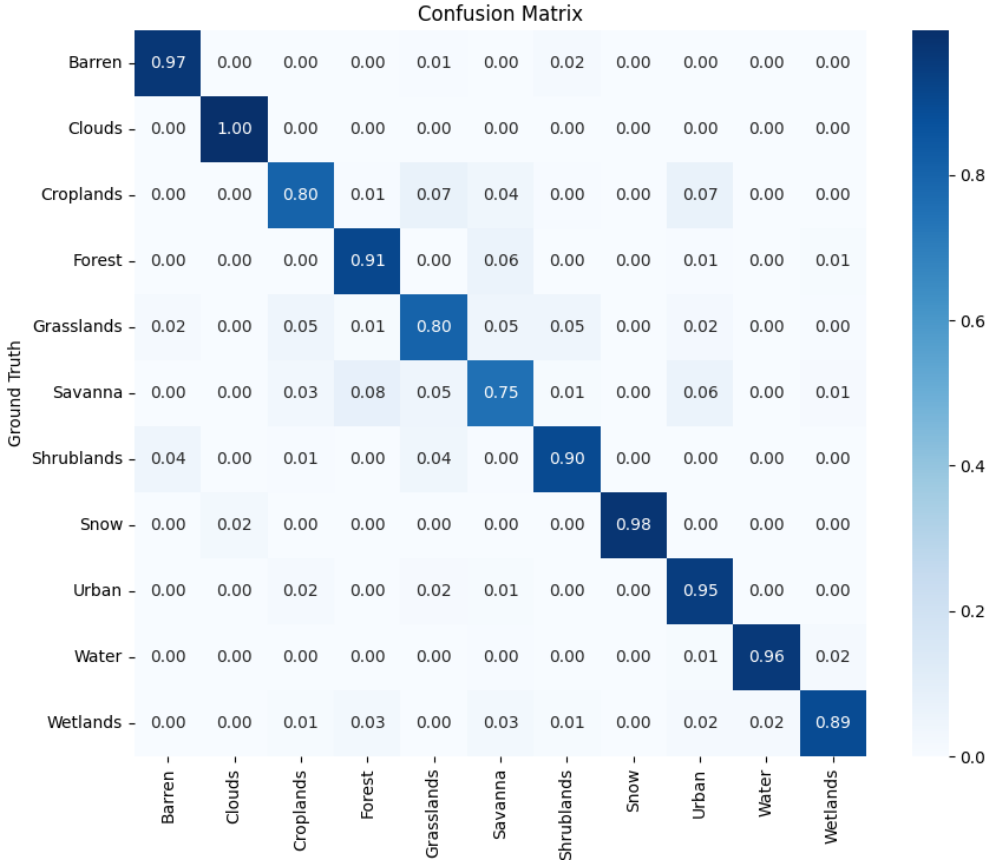


Fig. 4. Confusion matrix results for the pre-launch model on a Sentinel-2 image classification task.

Further insight into classification model outputs on individual image tiles was gained through the application of saliency maps via gradient-weighted class activation mapping (GradCAM) [26], an explainable AI technique to aid machine learning operators in understanding what features the model uses to discern classes in classification tasks. In Figure 5 we show the results of using GradCAM on an image with the ground truth label ‘water’ to create a heatmap of which features the model identifies as predicting the presence of water. We can query the model to identify which features it assigns as urban. The two heatmaps are inversed as expected and the strength of the water saliency map matches the model output and the ground truth.

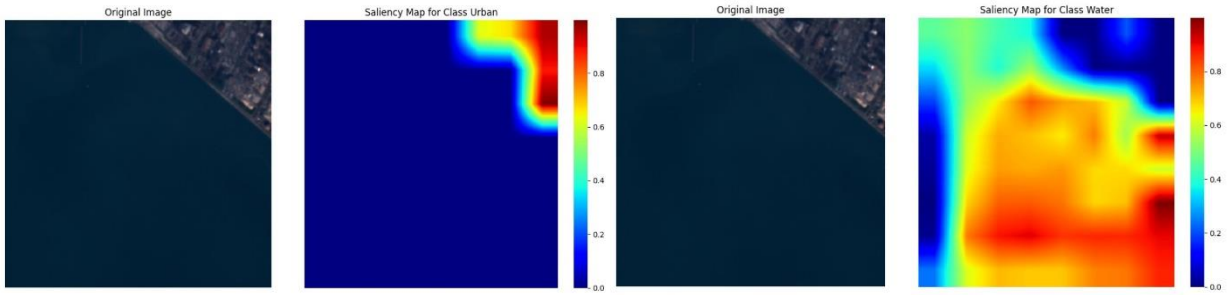


Fig. 5. Saliency maps for Urban class (left) and Water class (right) on the same input image.

*4.2 Simulated long term model performance and covariate shift*

To evaluate potential long term model performance and the efficacy of retraining the model over time to counteract the effects of time-dependent covariate shift, the SEN12MS dataset was split into 4 "learning blocks", each block consisting of data from different seasons of the year split into 80% train, 10% validation and 10% test data. Initially the model was trained on the first block only, it is then evaluated on the first block and the second block (Block 1, Table 1). The model is then refined using the training data of the second block and is evaluated again on the second block's and third block's test data (Block 2, Table 1). This is then repeated for the other two blocks, with the last one not possessing a "future" block for retraining.

Table 1. Accuracy results on multiple long-term blocks of imagery with models updated by MLOps.

| Model Update Simulation (MobileNetv3) | | |
|---|---|---|
| Block | Test Accuracy on Current Block | Test Accuracy on Next Block |
| 1 | 76.24% | 77.56% |
| 2 | 78.73% | 77.35% |
| 3 | 81.14% | 80.50% |
| 4 | 81.7% | |

Table 1 shows that model performance is improving over time as it is trained on the new data of the next block. As expected, the test accuracy of the next block is decreasing between block 1 and 2, the increase when moving to block 3 can likely be explained by the fact that the data used is not a true temporal dataset and that after completing the third data block the model has seen 75% of the training data and was able to generalize more broadly.

## 5. Conclusions

We have developed and tested an MLOps pipeline ground segment to support a flight segment that contains multiple machine learning flight payloads that will operate for at least one year on a 6U Cubesat. Using Sentinel-2 data prior to launch we trained an image classification model and deployed it to the onboard flight computer after rigorous HITL testing and V&V of the model performance via a combination of clustering, saliency maps, class-based metrics such as confusion matrices and an experiment that partitioned the training data into simulated temporal blocks to measure the MLOps pipeline's ability to compensate for covariate shift effects which could lead to model degradation. Launch of Mission Persistence is scheduled for June 2025 and during on-orbit commissioning and operations we will compare the performance differences between neural networks that are updated via the MLOps pipeline retraining on newly acquired data and static benchmarks models whose weights are frozen after the commissioning phase.

The MLOps framework here has potential to evolve into future space system architectures and novel observing strategies, such as the use of space-based data centres to host the training hardware [27] or the potential for in-orbit training on the satellite itself [28]. Beyond low-earth orbit, such as autonomous operations in deep space where

training data may not exist until operations commence, MLOps can be a key enabler of rapidly increasing deep learning model performance and perception and autonomy capabilities during mission operations. The move towards increased onboard autonomy [29] creates new mission profiles in NOS EO systems, space biology self-driving labs [30], and planetary science [19]. Future work will attempt to quantify the impact of MLOps in addressing covariate shift in operations and to extend its applicability to other types of domain shift to increase the robustness, resiliency, and trustworthiness of onboard deep learning applications in space missions.

**Acknowledgements**

Mission Control acknowledges the support of the Canadian Space Agency [SIMA03P2]. The authors also extend their appreciation and thanks to Jonathan Barr, Siddharth Dave, and the team at Spire Global for their work to design and develop the Mission Persistence satellite.

**References**

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

[2] D. Sculley *et al.*, 'Hidden Technical Debt in Machine Learning Systems', *Proc. Neural Inf. Process. Syst. NIPS*, 2015.

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

[4] K. L. Hobbs *et al.*, 'Space Trusted Autonomy Readiness Levels', in *2023 IEEE Aerospace Conference*, Big Sky, MT, USA: IEEE, Mar. 2023, pp. 1–17. doi: 10.1109/AERO55745.2023.10115976.

[5] S. Chien *et al.*, 'An Autonomous Earth-Observing Sensorweb', *IEEE Intell. Syst.*, 2005.

[6] J. L. Moigne, 'Novel Observing Strategies For NASA Future Earth Science Missions', 2023.

[7] J. L. Moigne and M. Cole, 'Advanced Information Systems Technology (AIST) New Observing Strategies (NOS) Workshop', 2020.

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

[9] F. Schöttl, A. Spichtinger, J. Franquinet, and M. Langer, 'Real-Time On-Orbit Fire Detection on FOREST-2', in *IGARSS 2024*, Athens, Greece: IEEE, Jul. 2024. doi: 10.1109/IGARSS53475.2024.10641618.

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

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

[12] A. Marin *et al.*, 'Φ-Sat-2: Onboard AI Apps for Earth Observation', 2021.

[13] S. Lu *et al.*, 'Onboard AI for Fire Smoke Detection Using Hyperspectral Imagery: An Emulation for the Upcoming Kanyini Hyperscout-2 Mission', *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 17, pp. 9629–9640, 2024, doi: 10.1109/JSTARS.2024.3394574.

[14] J. Nalepa *et al.*, 'Towards On-Board Hyperspectral Satellite Image Segmentation: Understanding Robustness of Deep Learning through Simulating Acquisition Conditions', *Remote Sens.*, vol. 13, no. 8, p. 1532, Apr. 2021, doi: 10.3390/rs13081532.

[15] D. Rijlaarsdam *et al.*, 'The Next Era for Earth Observation Spacecraft: An Overview of CogniSAT-6', *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 18, pp. 2450–2463, 2025, doi: 10.1109/JSTARS.2024.3509734.

[16] S. Kumari and P. Singh, 'Deep learning for unsupervised domain adaptation in medical imaging: Recent advancements and future perspectives', Jul. 18, 2023, *arXiv*: arXiv:2308.01265. doi: 10.48550/arXiv.2308.01265.

[17] A. J. Macdonald *et al.*, 'Enabling Autonomy with a Deep Learning Framework for Planetary Exploration', in *ASCEND 2022*, AIAA, 2022, p. 4295. doi: https://doi.org/10.2514/6.2022-4295.

[18] L. Chavier *et al.*, 'Deploying Artificial Intelligence Capabilities by Hybridizing a Neural Network on a Satellite', presented at the ASCEND, AIAA, 2023. doi: https://doi.org/10.2514/6.2023-4718.

[19] M. Cross *et al.*, 'Enabling Autonomy and Operations for Lunar Surface Missions: An Overview of Demonstrated Capabilities', in *17th Symposium on Advanced Space Technologies in Robotics and Automation*, Leiden, Netherlands, Oct. 2023. Accessed: Apr. 22, 2024.

[20] M. Schmitt, L. H. Hughes, C. Qiu, and X. X. Zhu, 'SEN12MS – A CURATED DATASET OF GEOREFERENCED MULTI-SPECTRAL SENTINEL-1/2 IMAGERY FOR DEEP LEARNING AND DATA

FUSION', *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. IV-2/W7, pp. 153–160, Sep. 2019, doi: 10.5194/isprs-annals-IV-2-W7-153-2019.
[21] M. Schmitt and H. Lloyd, 'SEN12MS'. Jan. 14, 2019. doi: doi:10.14459/2019mp1474000.
[22] 'Copernicus Data Browser'. [Online]. Available: https://browser.dataspace.copernicus.eu
[23] 'Living Atlas Land Cover Classification'. [Online]. Available: https://livingatlas.arcgis.com/landcoverexplorer
[24] 'Sentinel-2 L1C', Sentinel Hub. Accessed: Apr. 04, 2025. [Online]. Available: https://docs.sentinel-hub.com/api/latest/data/sentinel-2-l1c/
[25] A. Howard *et al.*, 'Searching for MobileNetV3', Nov. 20, 2019, *arXiv*: arXiv:1905.02244. doi: 10.48550/arXiv.1905.02244.
[26] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, 'Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization', *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 336–359, Feb. 2020, doi: 10.1007/s11263-019-01228-7.
[27] M. Gumiela *et al.*, 'Benchmarking Space-Based Data Center Architectures', in *IGARSS 2023 - 2023 IEEE International Geoscience and Remote Sensing Symposium*, Pasadena, CA, USA: IEEE, Jul. 2023, pp. 4970–4973. doi: 10.1109/IGARSS52108.2023.10281701.
[28] V. Růžička *et al.*, 'Fast model inference and training on-board of Satellites', Jul. 17, 2023, *arXiv*: arXiv:2307.08700. Accessed: Jul. 31, 2023. [Online]. Available: http://arxiv.org/abs/2307.08700
[29] I. Nesnas, L. Fesq, and R. Volpe, 'Autonomy for Space Robots: Past, Present, and Future', *Curr. Robot. Rep.*, vol. 2, Sep. 2021, doi: 10.1007/s43154-021-00057-2.
[30] L. M. Sanders *et al.*, 'Beyond Low Earth Orbit: Biological Research, Artificial Intelligence, and Self-Driving Labs'.