Benchmarking Deep Learning Models on Myriad and Snapdragon Processors for Space Applications

Emily R. Dunkel [*], Jason Swope[†], Alberto Candela[‡], Lauren West [§], Steve A. Chien[¶], and Zaid Towfic [‖]
Jet Propulsion Laboratory, [**] California Institute of Technology, USA

Léonie Buckley [††], Juan Romero-Cañas [‡‡], Jose Luis Espinosa-Aranda [§§] and Elena Hervas-Martin [¶¶]
Ubotica Technologies, Ireland

Mark R. Fernandez [***]
Hewlett Packard Enterprise

Future space missions can benefit from processing imagery onboard to detect science events, create insights, and respond autonomously. One of the challenges to this mission concept is that traditional space flight computing has limited capabilities because it is derived from much older computing to ensure reliable performance in the extreme environments of space, particularly radiation. Modern Commercial Off The Shelf (COTS) processors, such as the Movidius Myriad X and the Qualcomm Snapdragon, provide significant improvements in small Size Weight and Power (SWaP) packaging and offer direct hardware acceleration for deep neural networks, although these processors are not radiation hardened. We deploy neural network models on these processors hosted by Hewlett Packard Enterprise's Spaceborne Computer-2 onboard the International Space Station (ISS). We find that the Myriad and Snapdragon DSP/AIP provide speed improvement over the Snapdragon CPU in all cases except single pixel networks (typically >10x for DSP/AIP). In addition, the discrepancy introduced through quantization and porting of our JPL models was usually quite low (<5%). Models are run multiple times, and memory checkers are deployed to test for radiation effects. To date, we have found no difference in output between ground and ISS runs, and no memory checker errors.

[*] Data Scientist, Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Dr, Mail Stop 158-242, Pasadena, CA 91109, USA, emily.dunkel@jpl.nasa.gov

[†] Data Scientist, Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Dr, Pasadena, CA 91109, USA

[‡] Data Scientist, Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Dr, Mail Stop T1721, Pasadena, CA 91109, USA

[§] JPL Student Intern, Harvey Mudd College, Claremont, CA 91711, USA

[¶] Engineering Fellow, Senior Research Scientist, and Group Supervisor, Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Dr, Mail Stop 301-221, Pasadena, CA 91109, USA

[‖] Group Supervisor, Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Dr, Mail Stop 161-260, Pasadena, CA 91109, USA

[**] (c) 2023 All rights reserved. Government sponsorship acknowledged. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

[††] Computer Engineer, Ubotica Technologies, Old Finglas Rd, Glasnevin, Dublin, Ireland

[‡‡] Software Engineer, Ubotica Technologies, Camino de Moledores s/n, Incubadora de Empresas, Local 0.32, 13005, Ciudad Real, Spain

[§§] Principal Engineer, Ubotica Technologies, Camino de Moledores s/n, Incubadora de Empresas, Local 0.32, 13005, Ciudad Real, Spain

[¶¶] Software Engineer, Ubotica Technologies, Camino de Moledores s/n, Incubadora de Empresas, Local 0.32, 13005, Ciudad Real, Spain

[***] Principal Investigator, Spaceborne Computer-2, Hewlett Packard Enterprise, 1701 E. Mossy Oaks Rd., Spring, TX 77389, USA

I. Introduction

Deep space missions have limited contact with ground operations teams, making it difficult to account for a dynamic environment. This is due, in part, to the finite number of Earth-based ground communication stations and geometric constraints. Surface missions are typically commanded daily or every few days, and orbiters are typically commanded only weekly. Onboard autonomy can mitigate this by enabling spacecraft to autonomously respond to a changing environment in between command cycles. However, traditional space flight hardware has very limited computational capabilities. A new generation of embedded processors, such as the Intel Movidius X [1], and the Qualcomm Snapdragon 855 [2], enable fast onboard inference by supporting neural networks directly in hardware [3]. This technology potentially offers more powerful onboard autonomy with edge processing.

We benchmark deep learning inference on Movidius Myriad X and Snapdragon processors on the ground, as well as onboard the ISS. Hosting of these processors onboard is enabled by Spaceborne Computer-2 (SBC-2) by Hewlett Packard Enterprise (HPE) [4]. Previously, these models have been deployed on the ground on laptop, desktop, or cluster computers only. Here, we port these models to the Myriad and Snapdragon edge processors, and run inference on the ground and onboard the ISS. Although the radiation on the ISS is not as great as onboard a satellite, the radiation is higher than on Earth. We run onboard inference with these models at various times to see if errors occur, and in addition, run algorithms specifically designed to find memory errors, to quantify potential radiation effects. Models benchmarked cover a range of the types of algorithms we may want to run onboard a spacecraft in the future.

The ISS deployment is a step towards running such models operationally on a satellite, lander or rover. This would enable onboard data analysis, targeted downloads, commanding of space assets, and onboard science interpretation.

II. Experimental Setup and ISS Deployment

Two Snapdragon 855 development boards (with radios disabled) and two Movidius Myriad X Processors were integrated with the Hewlett Packard Enterprise (HPE) Spaceborne Computer-2, which was launched on February 20th, 2021, as part of the Cygnus NG-15 resupply mission to the ISS. The Snapdragons receive the required 12-V DC power from a standard PCI slot via a 2 pin power cable and communications via a standard internal USB connection. The Myriad X processors are connected through a USB connection. There is no third party hardware involved in the integration, and the processors do not become space-grade when hosted in the HPE computer. Uplinks are possible periodically to load new software.

An additional two Snapdragon 855 boards and two Myriad X Processors were included in HPE's ground testbed.

We show our experimental setup in Figure 1. Jet Propulsion Laboratory (JPL) and Ubotica port the models to the edge processors, develop and test locally on the ground. JPL tests on HPE's Ground Testbed. JPL sends HPE a test harness script, and they run the tests on their Flight Testbed on the ground. Once the models have been verified, HPE uploads and deploys the models on SBC-2 onboard the ISS. Results are then sent back to JPL for interpretation. All

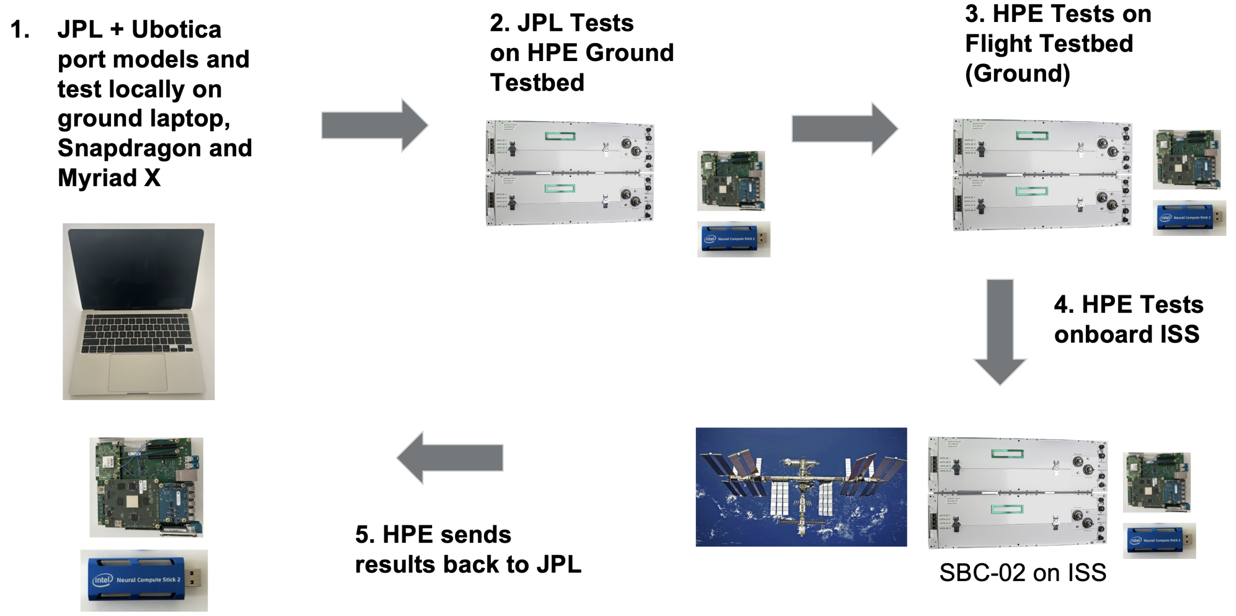

1. JPL + Ubotica port models and test locally on ground laptop, Snapdragon and Myriad X
2. JPL Tests on HPE Ground Testbed
3. HPE Tests on Flight Testbed (Ground)
4. HPE Tests onboard ISS
5. HPE sends results back to JPL
SBC-02 on ISS
intel Neural Compute Stick 2

Fig. 1 Experiment Setup. 1: JPL and Ubotica port the models and test locally on the ground. 2: JPL tests on HPE's Ground Testbed. 3: HPE tests on their Flight Testbed, which is located on the ground. 4: HPE uploads the models and data to the ISS and tests. 5. HPE sends results back to JPL for interpretation. SBC-2 imagery courtesy of HPE; ISS Image Credit: NASA.

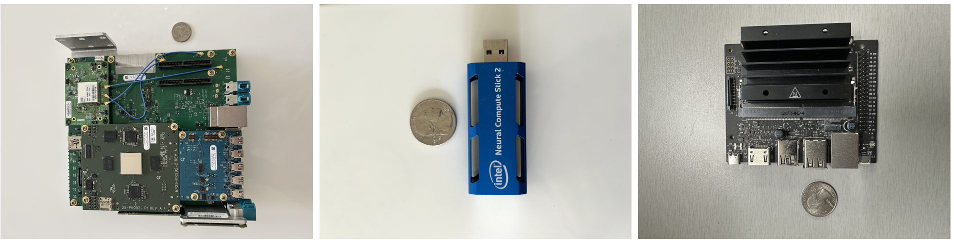
Snapdragon and Myriad X processors used in this experiment, are the same versions and do not have any radiation hardening. In addition, the same models and data are tested on all the processors, whether they are local, on a testbed, or on the ISS.

The Snapdragon and Myriad X processors on the ISS receive more radiation than on Earth; the ISS is in a low Earth orbit, with an apogee of 420 km, perigee of 380 km, and a 51.6 degree orbital inclination. The ISS does not go over the Earth's polar regions, where it would be subject to increased radiation due to reduced Van Allen Belt protection [5] [6]. In addition, the processors are shielded by the ISS itself.

Trained deep learning models are ported to formats that can be run on the Myriad X and/or the Snapdragon. Snapdragon deep learning models are ported using the Qualcomm Neural Processing Software Development Kit [7] to DLC (Deep Learning Container) format. We also test some scikit-learn [8] models ported to the Snapdragon (CPU only) using Python-for-Android by Kivy [9]. Myriad models are ported using OpenVINO [10] and the CogniSat-TK.

A. Processors

The Qualcomm Snapdragon 855 has multiple subsystems, which include a CPU, GPU, Compute Digital Signal Processor (DSP), and an AI Processor (AIP). The CPU subsystem is made up of a heterogeneous cluster of 8 ARM cores. One core operates at 2.8 GHz, three at 2.4 GHz, and four at 1.8 GHz. The Snapdragon also includes an Adreno GPU, which operates at 585 MHz and is geared for floating-point processing. In general, GPUs have been used widely



(a) Snapdragon 855 (b) Movidius Myriad X (c) Nvidia Jetson Nano

Fig. 2 Photos of the Snapdragon 855 development board, Movidius Myriad X chip, and Nvidia Jetson Nano board. We show a quarter (diameter of about 1 inch) in each photo for size reference.

for training and inference of neural networks, due to their parallelizable nature, and have allowed for development of larger and more accurate networks. However, the Snapdragon Hexagon DSP is even faster for inference than its GPU, due to its higher core speed and long vector length SIMD (Single Instruction Multiple Data) instructions for fixed-point computation. The DSP includes four cores operating at 1.2-1.3 GHz. The AIP adds accelerated computation for specific, commonly utilized, neural network functions. We compare results using the CPU, GPU, DSP, and Neural Processing Unit (NPU), which may include computation with the AIP; the NPU is the API that can be used to select the right component for a given task. The CPU and GPU support floating point numbers, while the DSP/NPU support fixed-point only and thus models must be quantized [2]. The Snapdragon Neural Processing Engine (NPE) Software Development Kit (SDK) [7] and Python-for-Android by Kivy [9] were used to develop our applications. The Snapdragons we benchmark run the Android operating system. A picture of the Snapdragon we test on locally is shown in Figure 2.

Snapdragon processors have been used in vehicles, drones, and even the Mars Ingenuity Helicopter and base station [11]. Experiments are being performed to assess the level of radiation that can be tolerated. Reference [12] discusses the results of experiments performed on Snapdragon processors at the Texas A&M University's Accelerator Laboratory. The boards were irradiated while running our Mars MSL model (Section III.A.2) and memory checkers (Section V.B). The Single Event Functional Interrupt (SEFI) was characterized as a function of beam energy. As long as the processors remained running and did not crash from the radiation, there was no deviation in output found in the experiment; that is all inference outputs were as expected and the memory checkers found no errors.

The Intel Movidius Myriad X Vision Processing Unit (VPU) features two Neural Compute Engines (NCEs). The NCE is a dedicated hardware block for accelerating neural network inference. The Myriad X VPU also includes 16 SHAVE vector processors - VLIW SIMD cores that can be used to execute inference operations and computer vision algorithms. These two NCEs can be run in parallel, meaning that the inference time when utilising both NCE's can be halved. Note that unless specified, the execution time provided for the Myriad X includes the time required to transfer from the host machine to the Myriad, and takes into account the benefit of using both NCE's. If only the inference execution time on the Myriad X is used, the benefit of using both NCE's will not be taken into account, that is, the time reported will be for a single NCE. The VPU is programmable using Ubotica's CogniSat-TK™ [13]. Half precision floating point is supported. See Figure 2 for an image of our local hardware.

A previous generation Myriad VPU, the Myriad 2, flew on the on the Φ-Sat-1 satellite, a CubeSat mission from the European Space Agency [3], which launched in September 2020 and completed operation in December 2021 when the mission objectives were achieved. The extent to which the Myriad 2 is affected by radiation was analyzed as part of the Φ-Sat-1 project in order to thoroughly characterise its radiation response and to ensure reliable operation in orbit [14]. Myriad characterisation in heavy ions and protons was performed, as well as characterisation for Total Ionizing Dose (TID). This extensive radiation characterisation campaign provides insight into the expected performance in orbit.

We compare results from the Snapdragon CPU, GPU, DSP, and NPU, the Myriad X, and a test laptop, all on the ground. When available, and for the majority of our models, we also compare with an NVIDIA Jetson Nano, which has been developed for embedded GPU and AI intensive applications [15]. The NVIDIA Jetson Nano features a 128-core NVIDIA Maxwell™ GPU and Quad-core ARM® A57 CPU that operates at 1.43 GHz. The Jetson Nano power consumption ranges from 5-10W [16]. The test laptop is a 2019 MacBook Pro, with a 2.4 GHz and 8-Core Intel i9 processor, operating at 3.1W idle and 94W max (includes monitor and other externals), running Ubuntu 18.04 in a docker container.

We also looked into the Rad750 processor, which has been used on many prior missions, including the Mars Science Laboratory (MSL) and Mars 2020 Mars Rover Missions [17]. We were not able to run deep learning on the RAD750, however, due to lack of common deep learning framework support (e.g. TensorFlow [18]). We were also not able to benchmark with the SBC-2 NVIDIA Tesla T4 GPU [19] due to incompatibility between CUDA driver/libraries with our deep learning frameworks. We used TensorFlow v1.6 and Caffe [20] libraries, which were supported by our version of the Snapdragon SDK (version 1.39.1) [21], but which are incompatible with the newer CUDA drivers. We do, however, provide benchmarks with a modern MacBook laptop and the Jetson Nano, a common embedded AI processor.

B. Methods

We benchmark a variety of models developed at JPL of the type we may want to run onboard in the future. In addition, we test some standard pre-trained classification models that come with the Keras package [22], since fine-tuning pre-trained networks is a common approach, especially when training data are limited. These models are first ported to and benchmarked on the Snapragon CPU, GPU, DSP, NPU, and Myriad X on the ground. When available, we also include Jetson Nano results. The models we benchmark range from image classification to image segmentation and spectral unmixing. The datasets we used are the test sets used by the model developers.

Our goals are to see which models can be ported, analyze which types of models receive a speedup from running on these edge processors, measure any errors that the model porting may cause, and assess the effort to port/deploy for future applications. Not all neural network layers are supported by the Myriad X or Snapdragon Software Development kits (SDKs). At the time we performed these experiments, the Snapdragon SDK was not compatible with TensorFlow 2 or PyTorch. The models that we benchmark were originally in or compatible with TensorFlow 1 or Caffe Deep Learning Framework.

In addition to these ground-based tests, we also test onboard the ISS to see if there are any effects from radiation. The radiation environment on the ISS is not as great as on a satellite but is greater than on Earth. Finally, we run memory checkers onboard the ISS to quantify possible radiation effects.

For each experiment, we report the run times as well as the error from porting the model (the difference in output between a reference MacBook run and run on the edge processors). We call this error the quantization discrepancy. Models that are run on the Snapragon DSP/NPU must be quantized (fixed point), and on the Myriad X must be transformed to half precision floating point. This leads to a discrepancy in model output. Models are quantized to fixed point for the Snapdragon using a separate validation set, and results are reported on the held-out test set. Errors may also occur from porting to the Snapdragon GPU due to floating point operation differences from the CPU.

For a classification algorithm, for example, we report the number of predicted classes that deviate between a MacBook run versus a edge processor run. This is not the number of mis-classified objects, but is rather the difference in classification output between a MacBook and edge processor run. Also, note that the inference output is exactly the same with each run of the model; the quantization errors are the same no matter how many ground runs we do.

Each of our models have been run on the ground as well as onboard the ISS.

III. Benchmarking Results

We benchmark models trained on Mars and Earth imagery, as well as some standard classification models included with the Keras package. As described in Section II, we take these trained models and port them to the Snapdragon using the Snapdragon NPE Software Development Kit [7] or Python-for-Android by Kivy [9]. Myriad models are ported using OpenVINO [10] and the CogniSat-TK. All results are for inference runs on a test set.

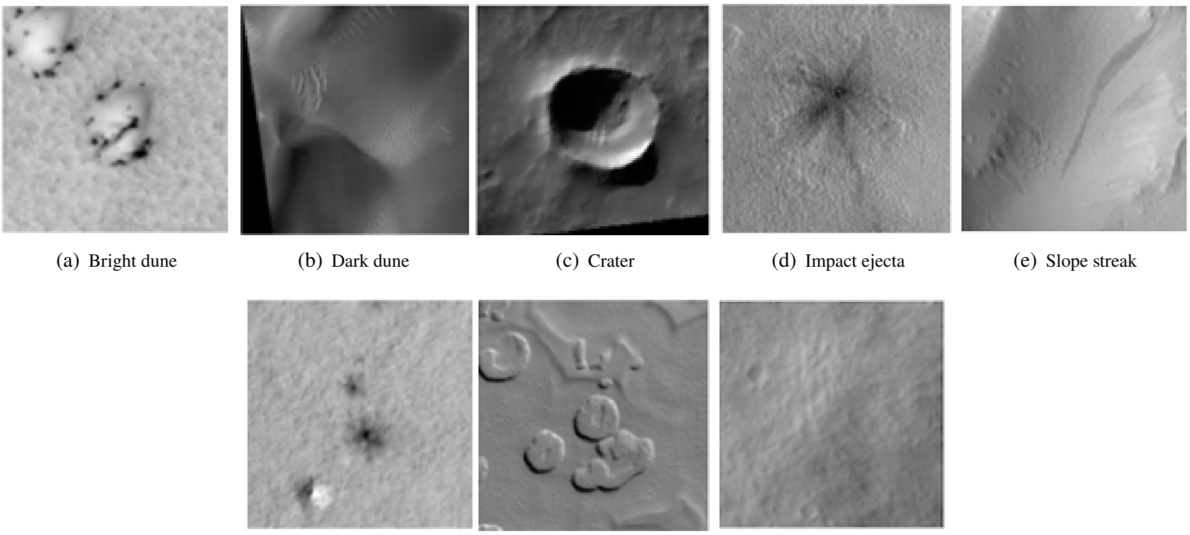
Inference results obtained were the same whether run on the ground or onboard the ISS, so we did not see radiation effects here. We also run memory checkers onboard the ISS as an additional test for radiation effects, and to date, have not found any errors.

In Section III.C we summarize and compare benchmarking results over all the models.

A. Mars Imagery Models

In this section, we discuss classification models trained using imagery from the Mars Reconnaissance Orbiter (MRO) and the Mars Science Laboratory (MSL) rover's science cameras. We also benchmark an image segmentation model trained on the MSL rover's navigation cameras.

A subset of the results of MRO and MSL navigation imagery models were shown in [23]. In addition, partial results



(a) Bright dune (b) Dark dune (c) Crater (d) Impact ejecta (e) Slope streak

(f) Spider (g) Swiss cheese (h) Other

Fig. 3 Example imagery for each HiRISENet classfrom the HiRISE instrument onboard MRO [25]. Imagery is publicly available on the PDS Image Atlas (https://pds-imaging.jpl.nasa.gov/search/).

for MSL science imagery models were included in [24] (non-archival, link included in the bibliography). Here, we include results from the Jetson Nano CPU and GPU for all Mars models, and give any errors from ISS deployment.

The Mars data is publicly available on NASA's Planetary Data System (PDS) at https://pds-imaging.jpl.nasa.gov/. In addition, the Mars HiRISENet and MSLNet models are planned to be available on the PDS website by end of 2023.

For our classification models, we find that the Snapdragon DSP/NPU and Myriad X have only small quantization discrepancies, with up to 100x speed improvement over the Jetson Nano CPU. Our image segmentation model, however, shows high quantization discrepancy for the Snapdragon DSP and less relative speed improvement (10x) over the Jetson Nano CPU. The Snapdragon SDK was not able to pre-quantize our DeepLabV3 image segmentation model, so our runtime quantization was likely the cause for greater discrepancy.

1. Mars HiRISENet Classifier

The NASA PDS maintains archives of data collected by NASA missions, and provides access to millions of images of planets, moons, comets, and other bodies to the general public. This includes images from the High Resolution Imaging Experiment (HiRISE) instrument onboard the Mars Reconnaissance Orbiter (MRO). Users can interactively search these images for classes of interest using the PDS Image Atlas, which uses the predictions from HiRISENet (https://pds-imaging.jpl.nasa.gov/search/).

Imagery from MRO is used to study Martian surface features. HiRISENet is trained on classes that include dunes, craters, the debris from impact or volcanic eruptions (impact ejecta), and features formed by sublimating CO2 (spider and swiss cheese) [25]. Examples of these classes are shown in Figure 3. HiRISENet was built using transfer learning

Table 1 Mars HiRISE Classifier Benchmarks

Processor	Quantization Discrepancy	Inference Time	Energy
MacBook Reference	-	56.9 ms	2.3 J*
Snapdragon CPU	0	87.8 ms	0.5 J
Snapdragon GPU	1 (0.1%)	16.3 ms	0.051 J
Snapdragon DSP	15 (0.8%)	7.6 ms	0.016 J
Snapdragon NPU	15 (0.8%)	7.6 ms	0.014 J
Myriad X	2 (0.1%)	16.2 ms	0.032 J
Jetson Nano CPU	0	1,069 ms	5.3-10.7 J
Jetson Nano GPU	0	234 ms	1.2-2.3 J

from AlexNet [26] using Caffe. Test images were 227x227 pixels and in grayscale.

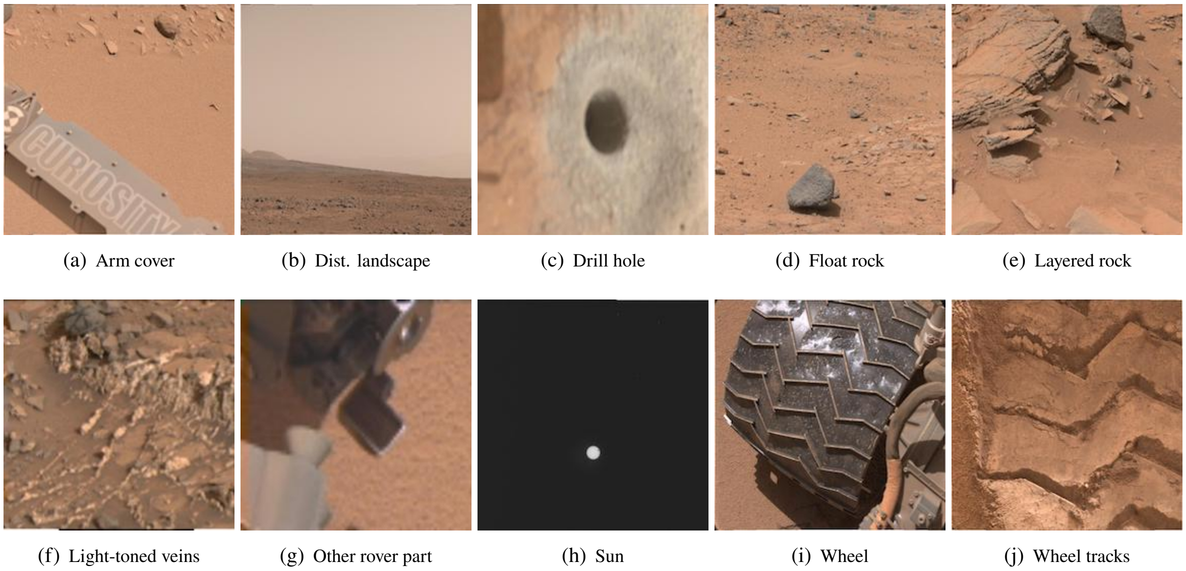
Models that are run on the Snapdragon DSP/NPU must be quantized (fixed point), and on the Myriad X, must be transformed to half precision floating point, which can lead to a discrepancy with the predicted class. Models are quantized using a separate validation dataset and quantization discrepancy errors are reported on a held-out test set. Table 1 shows quantization discrepancy, inference time, and power relative to a run on the test laptop: MacOS 2019, 2.4GHz, 8-core, 3.1W idle, 94W max *. Inference was performed on 1,793 test images; we show the number of images that have predicted class values different than the reference laptop, as well as the percentages out of the 1,793 image test set. Inference time and energy consumption shown are per image. On the test laptop, the time reported is walltime. The Snapdragon DSP/NPU and Myriad X have only small quantization discrepancy, with up to 10x speed improvement over the Snapdragon CPU, and over 100x speed improvement over the Jetson Nano CPU. To put the quantization discrepancy in context here, HiRISENet has 92.8% test set accuracy [25], so the quantization error is a small fraction of the model error. Also, note that inference gives the exact same results each time, so the quantization error stays the same with each run.

To measure energy consumption on the Snapdragon 855 development board [2], the Snapdragon Profiler tool [27] was used to collect power utilization over time. The tool logged power utilization as well as inference runtime while monitoring the subsystem on which the algorithm was performed. In order to compute the energy expended per inference, we multiplied the run-time per image (measured in milliseconds) by the power utilization at the time the image was analyzed. The product of which is utilized energy for inference per image, converted to Joules.

To measure the energy consumption on the Myriad X, the device is connected to a USB 2.0 port through the YZXStudio USB meter ZY1270. Each model that was benchmarked was compiled to run on 12 of Myriad X's available vector cores.

The Jetson Nano uses between 5W to 10W [16]. We were not able to measure the energy consumption directly, so we give a range determined from the timing and device wattage range.

*This energy figure includes display, keyboard and other laptop elements


(a) Arm cover (b) Dist. landscape (c) Drill hole (d) Float rock (e) Layered rock

(f) Light-toned veins (g) Other rover part (h) Sun (i) Wheel (j) Wheel tracks

Fig. 4 Example imagery from MSL Curiosity Rover's science cameras showing some example classes [25]. Imagery is publicly available on the PDS Image Atlas (https://pds-imaging.jpl.nasa.gov/search/).

We see that the energy consumption by the Snapdragon NPU, DSP, and Myriad X are orders of magnitude lower than the Jetson Nano and laptop reference. The Snapdragon GPU uses an order of magnitude less power than the Snapdragon CPU for the same computation.

The Snapdragon HiRISENet model has been run for 9 iterations, and the Myriad model has been run for two iterations at various times on the ISS. With each iteration, the models are run a total of 3 times. No differences have been found between ground and ISS runs of these classifiers.

In the future, a similar model could enable data analysis and science interpretation onboard an orbiter.

2. Mars MSLNet Classifiers

The NASA PDS Image Atlas also maintains archives of images from the Mars Science Laboratory (MSL) Curiosity rover's science cameras, as well as class predictions on these images from MSLNet. Users can interactively search these images for classes they are interested in. Specifically, MSLNet classifies images collected by the Mast Camera (Mastcam) and the Mars Hand Lens Imager (MAHLI) instruments mounted on the MSL Curiosity rover. Mastcam is a two instrument suite with left and right-eye cameras, and MAHLI is a single focusable camera located at the end of the rover's robotic arm.

MSLNet is actually made up of two networks: MSLNet1 and MSLNet2. MSLNet1 is trained on 19 classes, including float rock, light-toned veins, sun, wheel, and wheel tracks [25]. If MSLNet1 predicts "other rover parts", the image will be passed through MSLNet2 for finer grained classification of 24 possible classes [28]. See Figure 4 for some example images with their class.

Currently, these classifiers are used only on the ground by the PDS Image Atlas, but running these classifiers directly onboard the rover could improve data collection and enable autonomous tasking.

Table 2 Mars MSL1 and MSL2 Classifier Benchmarks

	Quantization Discrepancy		Inference Time	
Processor	MSL1	MSL2	MSL1	MSL2
MacBook Reference	-	-	65.4 ms	69.1 ms
Snapdragon CPU	0	0	86.6 ms	81.6 ms
Snapdragon GPU	1 (0.2%)	1 (0.2%)	16.2 ms	16.2 ms
Snapdragon DSP	15 (2.5%)	27 (2.1%)	7.6 ms	7.6 ms
Snapdragon NPU	15 (2.5%)	27 (2.1%)	7.6 ms	7.6 ms
Myriad X	3 (0.5%)	7 (0.5%)	16.1 ms	16.1 ms
Jetson Nano CPU	2 (0.33%)	0	1,122 ms	1,109 ms
Jetson Nano GPU	2 (0.33%)	1 (0.1%)	286 ms	242 ms

MSLNets were built with transfer learning from AlexNet [26] using Caffe, similar to our Mars HiRISE model. Test images were 227x227 pixels and RGB. Models that are run on the Snapdragon DSP/NPU must be quantized (fixed point), and on the Myriad X must be transformed to half precision floating point, which can lead to a quantization discrepancy.

Benchmarking results are similar for both classifiers, as they have the same model structure. Table 2 shows quantization discrepancy and timing on a test set of 602 images for MSLNet1 and 1,305 testset images for MSLNet2, relative to a run on the test Mac laptop. We show the number (and percentage) of test set images with class predictions that do not match the reference run. Inference time is per image. On the test laptop, the time reported is walltime.

The Snapdragon GPU has 5x speed improvement over the Snapdragon CPU, and the DSP/NPU are 2x faster than the GPU. The Myriad X speed is similar to the Snapdragon GPU. The Jetson Nano processors are more than an order of magnitude slower than the Snapdragon DSP/NPU and Myriad X. Quantization discrepancies were low for all processors.

Snapdragon models have been run for 9 iterations on the ISS, 3 times for each iteration, for a total of 27 runs. Myriad X models have been run for two iterations on the ISS, 3 times for each iteration, for a total of 6 runs. No differences have been found between ground and ISS runs of these classifiers.

3. Mars NavCam Image Segmentation

This model was trained using data from the MSL rover's navigation cameras (NavCam) [29], instead of its science cameras like MSLNet. In addition, the NavCam model does image segmentation; instead of predicting a single classification *per image*, it predicts a single classification *per pixel*. This model was developed to support MSL Rover Planners for hazard assessment and slip analysis, but could also be run onboard to improve rover safety. The model was built using TensorFlow 1 and a DeepLabv3 architecture [30].

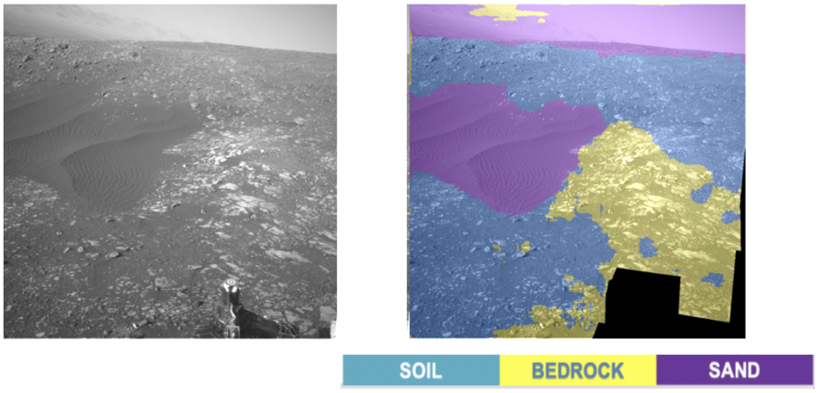


Fig. 5 Mars MSL NavCam Example Imagery and Label. Imagery is publicly available on the PDS Image Atlas (https://pds-imaging.jpl.nasa.gov/search/).

Table 3 Mars NavCam Image Segmentation Benchmarks

Processor	Quantization Discrepancy	Inference Time
MacBook Reference	-	1,886 ms
Snapdragon CPU	0.0%	6,235 ms
Snapdragon GPU	0.4%	2,233 ms
Snapdragon DSP	9.3%	192 ms
Snapdragon NPU	n/a	n/a
Myriad X	n/a	n/a
Jetson Nano CPU	0.0%	7,110 ms
Jetson Nano GPU	0.0%	2,717 ms

Figure 5 shows an example NavCam image with its corresponding label. Images are 513x513 pixels, and there are four possible class labels: soil, bedrock, sand, and big rock.

Table 3 shows the average quantization discrepancy and inference times per image (there were 322 test images, and each image was 513x513 pixels) for each processor. Models run on the Snapdragon DSP are quantized, and models run on the other processors are un-quantized. The quantization discrepancy column shows the percentage of pixels incorrectly labeled on average for each image. This quantization discrepancy on the Snapdragon DSP is relatively high. The system was unable to use a pre-quantized model and used run-time quantization, which increases the network initialization time, peak memory usage, and model file size, as well as affecting quantization discrepancy. We do not show Myriad X results, as the model had incompatible layers. The Jetson Nano CPU and GPU runtime and performance were similar to the Snapdragon CPU and GPU respectively. Run times on the Snapdragon DSP were more than 10x faster than on the Snapdragon GPU or Jetson Nano GPU, although this comes at the price of high quantization discrepancy.

The Snapdragon models have been run for 7 iterations, 3 times for each iteration, for a total of 21 runs onboard the ISS. No differences have been found between ground and ISS runs of these image segmentation models.

B. Earth Imagery Models

Going back to Earth, we look at a variety of models ranging from image segmentation to spectral super resolution and unmixing. Models include a UAVSAR segmentation model for flood mapping, single pixel cloud classifiers, a model to predict fractional amounts of coral and algae, and a model that predicts higher resolution spectral data from lower resolution data.

A subset of the results from the UAVSAR model were shown in [23] [31], a portion of the super resolution results in Reference [23], and some of the spectral unmixing work in non-archival Reference [24]. Here, we go into more detail on these models, and add benchmarks on the Myriad X for super resolution and spectral unmixing.

We find that single pixel models do not benefit from the embedded hardware, although the larger models have significant speed-up. The hardware is optimized for larger network structures. Thus, we do not recommend using the embedded hardware for single pixel models.

We also compare the run time of various single pixel machine learning models using the Python-for-Android routine by Kivy [9], which allows porting of standard scikit-learn models [8] directly for the Snapdragon CPU. This is to demonstrate the ability to run Python code directly on the Snapdragon CPU. We find, however, that the Snapdragon CPU offers little to no improvement over the Jetson Nano.

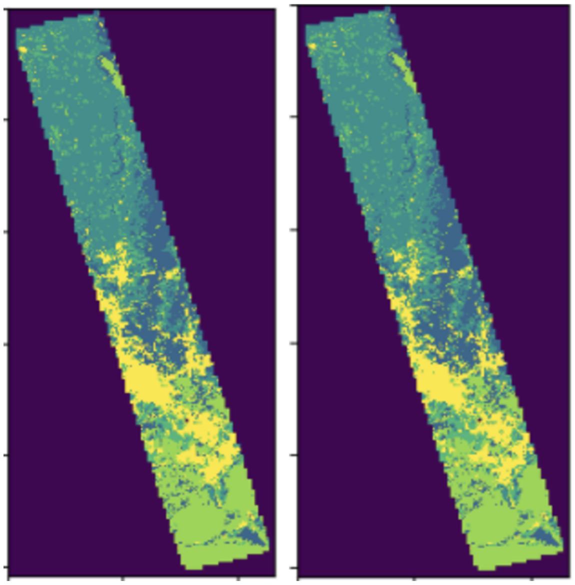
Finally, we benchmark a set of models that come standard with the Keras package [22]. We port these models to formats that can be run on the Snapdragon and Myriad X processors. Transfer learning from these pre-trained models is a common approach used when training data are limited, thus this may help with model selection for use with edge processing.

1. UAVSAR Model for Flood Mapping

This model was trained to perform pixel-wise classification of UAV polarimetric L-band SAR imagery, to predict areas that have been flooded. The model was trained on imagery of Houston, TX, USA, after flooding by Hurricane Harvey [32]. The net architecture follows a UNET [33] structure and was implemented in TensorFlow. In Figure 6 we show example output from a run on the Myriad X compared with the reference laptop; results are nearly indistinguishable. The model outputs 6 possible pixel-wise classes: unknown, flooded vegetation, non-flooded vegetation, bare ground, open water, and urban. To come up with a final classification output, classes are compressed into simply two classes: flooded or non-flooded.

In Table 4, we show benchmarking results for full classification and binary classification on the Snapdragon, Myriad X, and reference laptop (we do not have Jetson Nano results for this model). Image patches are 64x64 pixels. A run time of 1.3 image patches/second is needed to meet real time, and this is met by all platforms. All quantization discrepancies are small.

In the future, this model could be run onboard a UAV for onboard alert generation or surveillance.



(a) Reference (b) Myriad X Results

Fig. 6 SAR Image Segmentation Results. Light green denotes open water, dark blue denotes flooded vegetation, and remaining colors denote non-flooded regions.

Table 4 SAR Image Segmentation Benchmarks: 64x64 pixel image patches

Processor	Full Class. Discrepancy	Binary Class. Discrepancy	Image Patches / Second
MacBook Reference	-	-	25
Snapdragon CPU	3	1	20
Snapdragon GPU	4	1	162
Snapdragon DSP	87,689 (0.74%)	44,844 (0.38%)	391
Snapdragon NPU	87,689 (0.74%)	44,844 (0.38%)	391
Myriad X	184,820 (1.56%)	82,519 (0.70%)	167

The Snapdragon model has been run for 7 iterations, 3 times for each iteration, for a total of 21 runs onboard the ISS. The Myriad X model has been run for 3 iterations, 3 times each. No differences have been found between ground and ISS runs of these models.

2. SMICES Single Pixel Cloud Classifier

In this section, we benchmark single pixel cloud classifiers built with scikit-learn [8]. We compare a single pixel neural network model (a multi-layer perceptron model) with traditional classifiers on the Snapdragon CPU, Jetson Nano CPU, and MacBook Reference. We do not show results on the Myriad X since scikit-learn models cannot be ported with the Ubotica CogniSat-TK.

For the Snapdragon, we port these scikit-learn models using the Python-for-Android routine by Kivy [9]. This is different than all other deep learning models discussed in this paper, which use the Qualcomm Neural Processing SDK

Table 5 SMICES Classifier Run Times

Processor	RDF Classifier	MLP Classifier	Bayes Classifier
MacBook Reference	2.0 ms	1.6 ms	0.3 ms
Snapdragon CPU	2.5 ms	3.0 ms	1.5 ms
Snapdragon GPU	n/a	n/a	n/a
Snapdragon DSP	n/a	n/a	n/a
Snapdragon NPU	n/a	n/a	n/a
Myriad X	n/a	n/a	n/a
Jetson Nano CPU	2.5 ms	5.1 ms	1.5 ms

to port models to DLC format; the Qualcomm SDK allows for more flexibility and more complex network structures, and allows us to run on the Snapdragon GPU, DSP, and AIP as well as the CPU. Here, we simply demonstrate that scikit-learn models can be ported directly for use on the Snapdragon CPU. Future work involves comparison of the multi-layer perception scikit-learn model with a similar model in DLC format.

SMICES is an instrument concept for a "smart" deep convective storm hunting radar [34] [35] [36]. A look-ahead radiometer acquires data to detect deep convective ice storms and radar is used to study detected storms in greater detail. The SMICES machine learning application [37] classifies simulated radiometer data into five separate cloud types to identify the location of deep convective storms. The application runs a random decision forest (RDF), multi-layer perceptron (MLP), and naïve Bayes Gaussian classifier over 198,016 pixels with 8 bands of radiance. Each classifier is run on the Snapdragon CPU in a single threaded python application and compared with performance on a reference laptop and Jetson Nano.

Run times per pixel are shown in Table 5. For RDF and Bayes, the runtimes for the Snapdragon CPU are equal to the Jetson Nano CPU. The MLP, however, runs 1.7x faster on the Snapdragon.

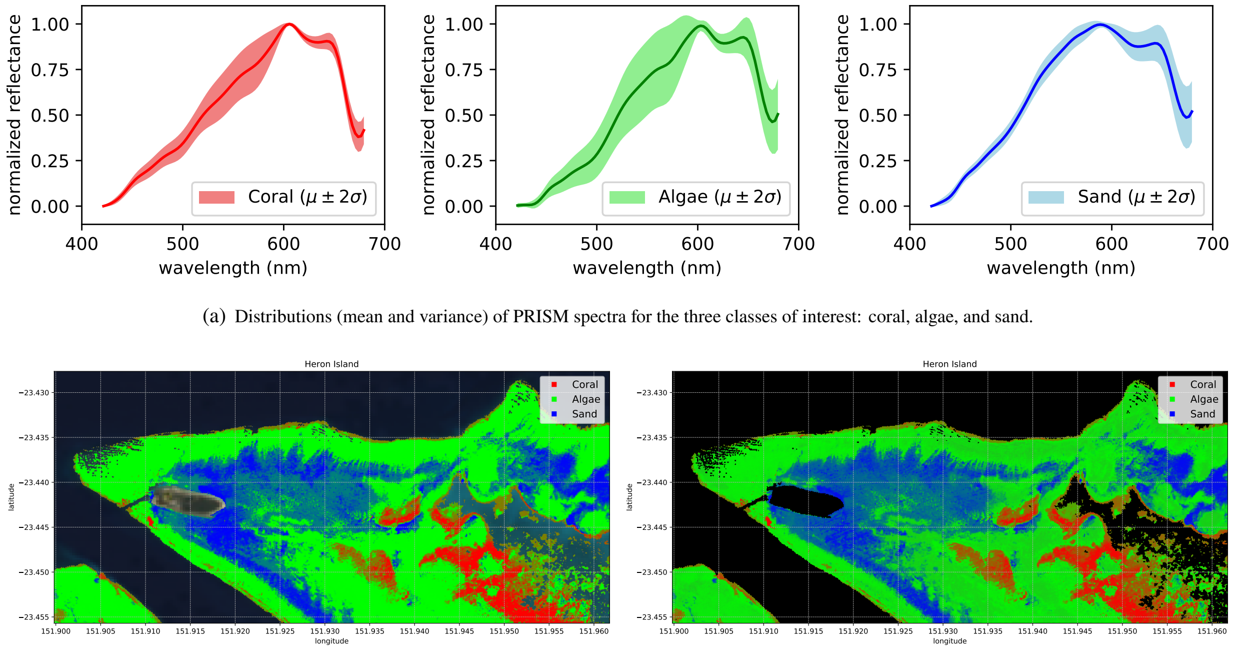
The Snapdragon SMICES classifiers have been run 9 times on the ISS Snapdragon, and we have found no differences between ground and ISS runs.

3. Spectral Unmixing

Earth and planetary sciences often rely upon the analysis of spectroscopic data. Measured signals are called spectra and contain recognizable features or patterns that can be used for composition analysis since different materials reflect, emit, or absorb energy in unique ways throughout the electromagnetic spectrum.

This model addresses spectral unmixing, an approach for estimating the proportions or fractional abundances of at least two components in each spectrum (e.g., 60% material A and 40% material B). Unmixing is more general than conventional classification as it models mixtures of classes, as opposed to a single class (e.g. simply all material A or all material B).

We benchmark the Deep Conditional Dirichlet Model (DCDM) [38], which is a probabilistic deep learning model



(a) Distributions (mean and variance) of PRISM spectra for the three classes of interest: coral, algae, and sand.

(b) Coral reef map from the CORAL mission, which serves as the ground truth.

(c) Coral reef map as predicted by the DCGM (93.6% accuracy).

Fig. 7 Benthic cover abundance maps of Heron Island as predicted by the DCDM using PRISM spectra [38].

for learning mixtures of classes. It can model both linear (non-intimate, i.e. that the signal is the weighted sum by abundance of the each element signature), and nonlinear (intimate) mixtures. This method treats each pixel individually, without looking at its surrounding neighbors. As a probabilistic method, it is robust to noise and also models uncertainty propagation in the data.

We demonstrate performance using airborne data from the NASA Earth Venture Suborbital-2 (EVS-2) Coral Reef Airborne Laboratory (CORAL) mission [39]. The CORAL mission focused on mapping three benthic cover classes: coral, algae, and sand. Data is from the NASA/JPL Portable Remote Imaging SpectroMeter (PRISM) (Figure 7). We employ two flight lines from Heron Island, Australia on 17 September 2016 and Kaneohe Bay in Oahu, Hawaii on 6 March 2017. We use benthic reflectance products since they provide invariance to water column properties. These products have a 420–680 nm spectral range and consist of 92 bands. The abundance maps were estimated and validated by the CORAL mission with photomosaics collected in the field [40]. The test dataset consists of 1,200 representative PRISM spectra together with their corresponding coral, algae, and sand fractional abundances.

Benchmarking results are shown in Table 6. We show performance on the Snapdragon subsystems, Myriad, and test laptop, running in batch mode (all pixels passed in at once). Runtime is per-pixel, and MacBook time is wall time. The quantization discrepancy is given as root mean square error (RMSE). For this model, the Snapdragon CPU is the faster than the other subsystems, most likely due to the small size of the model and single-pixel nature. The time provided for the Myriad X is the actual inference time and does not include the image transfer time. This model has been run three

Table 6 DCDM Benchmarks

Processor	Quantization Discrepancy (RMSE)	Inference Time
MacBook Reference	-	5 us
Snapdragon CPU	0	15 us
Snapdragon GPU	0	270 us
Snapdragon DSP	0.05	39 us
Snapdragon NPU	0.05	39 us
Myriad X	0.06	340 us

Table 7 Super Resolution Quantization Discrepancy: Missed Pixels out of 1,000 for Range of Tolerances

	Tolerance					
Processor	1e-6	1e-5	1e-4	1e-3	1e-2	1e-1
Snapdragon CPU	0	0	0	0	0	0
Snapdragon GPU	995	984	733	0	0	0
Snapdragon DSP	1000	1000	1000	1000	368	0
Snapdragon NPU	1000	1000	1000	1000	368	0
Myriad X	1000	1000	1000	956	0	0

times in one iteration on the ISS Snapdragon, and produced the same results as in the ground-based run.

4. Super-resolution

Earth and planetary scientists use high-resolution spectral measurements for rock and mineral identification. This data tends to be sparse, compared to lower resolution data. A Deep Conditional Gaussian Model was created to infer high-resolution measurements from low-resolution ones [41]. Specifically, it was trained to predict hyperspectral (high-resolution) measurements from the Airborne Visible Infrared Imaging Spectrometer Next Generation (AVIRIS-NG) using multispectral (low-resolution) Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) inputs. The data set consists of images covering Cuprite Hills, Nevada (Fig. 8); a well-studied region that is amenable to remote sensing and has a high mineralogical diversity [42]. This model used 97 AVIRIS-NG bands and 5 ASTER bands, both in the 2000-2500 nm spectral range.

In Table 7, we summarize the number of missed pixels (out of a 1,000 pixel testset) for a range of tolerances. There are a total of 97 predicted bands for each pixel and band values range from 0 to 1. For a given pixel, we compare each band with its reference value. If any of the band differences are greater than the listed tolerance, that pixel is considered to be missed.

In Table 8 we show run time per pixel. Inference on the Snapdragon DSP/AIP is twice as slow as running on the Snapdragon CPU (45 vs 21 ms per input). This is most likely due to the small size of the model and single-pixel nature. The time provided for the Myriad X is the actual inference time and does not include the image transfer time.

This model has been for two iterations on the ISS Snapdragon, three times each iteration, and produced the same

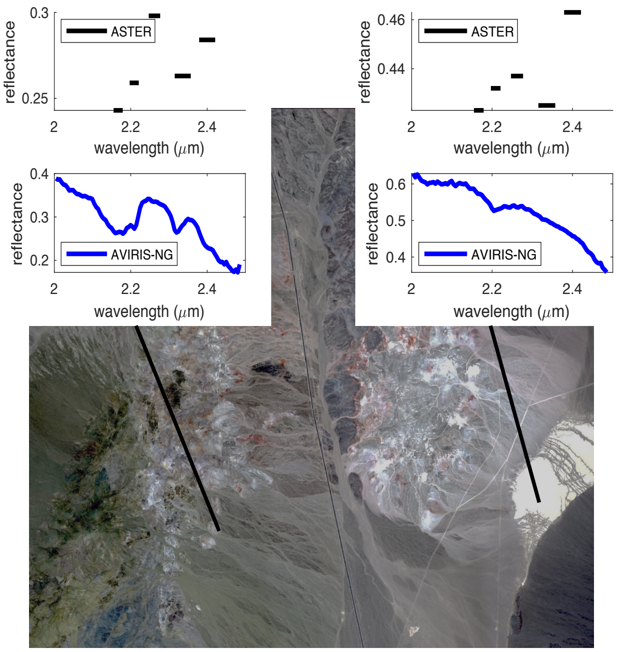


Fig. 8 The super-resolution data set consists of spectroscopic measurements of Cuprite Hills, Nevada. Each pixel in the image is associated with one low-resolution (ASTER) measurement and one high-resolution (AVIRIS-NG) observation. The goal of super-resolution is to infer high-resolution spectra from low-resolution data [41].

Table 8 Super Resolution Timing

Processor	Inference Time
MacBook Reference	150 us
Snapdragon CPU	21 us
Snapdragon GPU	276 us
Snapdragon DSP	45 us
Snapdragon NPU	46 us
Myriad X	290 us

results as in the ground-based run.

5. Standard Classification Models

Transfer learning from pre-trained models is often used for model development (and was used for the Mars MSL and HiRISE models we discussed earlier). We benchmark seven standard Keras [22] deep learning classification models on the Snapdragon and Myriad X, which may help with model selection for edge processing. This work addresses run time and quantization discrepancy only, not classification accuracy, which is also important in model selection. We compare Snapdragon and Myriad X results with a MacBook run to get quantization discrepancies. As with our prior models, this discrepancy is calculated using the model predictions obtained when running on the MacBook, not ground truth. Run times are compared with a MacBook, Jetson Nano CPU, and Jetson Nano GPU.

(a) Golf ball (b) Gas pump (c) Garbage truck (d) Church (e) Parachute

(f) Tench (g) Chain saw (h) Cassette player (i) English springer (j) French horn

Fig. 9 Examples of each class in Imagenette [44]. Images are available through Apache License 2.0 on https://github.com/fastai/imagenette.

All models were trained on ImageNet [43], which has 1,000 different classes, and contains internet imagery. For testing, we use Imagenette [44], which is a much smaller dataset, containing only 10 classes. Examples of each class are shown in Figure 9. Note that each model still predicts 1,000 possible classes, it is just that our test set contains only 10 classes. We randomly sample 10 images from each of these 10 Imagenette classes to create the test dataset of 100 images. We benchmark the following standard Keras models: MobileNet, Xception, InceptionV3, ResNet50, InceptionResNetV2, VGG16, and VGG19. The models were downloaded using the standard Tensorflow 1 Keras library, and the appropriate image pre-processing was applied for each of the networks.

In [23], we report benchmark results for the Snapdragon NPU only. Here we report on all Snapdragon systems, the Myriad X, Jetson Nano (run time only available), and our test laptop. In Table 9, we show quantization discrepancy on 100 test images from the Imagenette dataset compared with MacBook results. The quantization discrepancies for VGG16, VGG19, and ResNet50 on the Snapdragon NPU reported here are slightly better than what we show in [23], as we had swapped the red and blue channels of the input. We fix that pre-processing error here, although in both cases we used the same image format for all processors, which is why the comparison results are similar. VGG16, VGG19, and ResNet50 use the Caffe convention for normalization (red/blue swap and per channel subtraction of the ImageNet mean), and the other models use pixel scaling from -1 to 1. Not all models were able to be run on the Snapdragon DSP due to incompatible layers (marked as n/a), although all models were able to be run using the Snapdragon NPU.

As previously mentioned, the quantization discrepancy is calculated using the model predictions obtained when running on the MacBook, and not against the ground truth. The quantization discrepancy of these models tends to be much higher than the JPL models we benchmarked. This is not surprising, as the number of classes these standard models were trained on (1,000 classes) is much higher than the number of classes in our JPL models.

The quantization discrepancy of MobileNet is very high. The Snapdragon NPU has 100% classification discrepancy (out of 1,000 different possible predicted classes), meaning the predicted class for every image was incorrect, while the Myriad X had a smaller quantization discrepancy of 15%. The reason for these large quantization discrepancies for MobileNet may be due to the fact that the structure of the net gives greater fluctuation ranges, which may make it not easily quantize-able [45].

Table 9 Pre-Trained Models: Quantization Discrepancy

	MobileNet	Xception	InceptV3	ResNet50	InceptResNetV2	VGG16	VGG19
Snapdragon CPU	0	0	0	0	0	0	0
Snapdragon GPU	0	0	0	1%	0	0	0
Snapdragon DSP	n/a	n/a	n/a	4%	n/a	0	1%
Snapdragon NPU	100%	11%	4%	4%	16%	0	1%
Myriad X	15%	16%	11%	8%	9%	1%	1%

In Table 10, we show run times per 224 x 224 RGB image, along with the number of model parameters. Run times are all in ms. The MacBook times are walltime for running the Keras models. Runtimes on the Jetson CPU and GPU are much longer than for all other processors, regardless of model. Three models were not able to run on the Jetson Nano GPU, and are marked as n/a. On the Snapdragon NPU, ResNet50 has the fastest inference time, whereas the fastest inference time on the Myriad X was with MobileNet. Columns in Table 10 are ordered by the number of parameters (shown in the first row). Run time versus number of net parameters is processor dependent. Run time on the Snapdragon CPU, GPU, and Myriad X generally increase with number of net parameters, except for the Xception model. Run time on the Snapdragon NPU is not predicted by the number of net parameters. Snapdragon models were created from TensorFlow 1.6 models, while the Myriad X used TensorFlow 1.15.

C. Benchmarking Results Summary

In this section, we summarize the results of our benchmarking experiments.

The Snapdragon DSP or NPU provided inference speedups relative to the Snapdragon CPU in the majority of our experiments. The Snapdragon DSP/NPU is optimized to run neural network inference, but the speedup depends on the network architecture.

The Mars HiRISE, MSL1, and MSL2 classification models were all built using transfer learning from AlexNet,

Table 10 Pre-Trained Models: # Params and Run Time per Image (224 x 224 pixels, RGB) in ms

	MobileNet	Xception	InceptV3	ResNet50	InceptResNetV2	VGG16	VGG19
# Parameters	4e6	23e6	24e6	25e6	55e6	138e6	144e6
MacBook Ref	154	596	324	520	814	1,622	1,965
Snap CPU	200	505	368	388	872	1,192	1,496
Snap GPU	14	48	37	43	85	143	176
Snap DSP	n/a	n/a	n/a	10	n/a	27	31
Snap NPU	60	53	27	10	56	27	31
Myriad X	20	59	29	41	74	169	220
Jetson Nano CPU	46k	95k	225k	19k	339k	165k	180k
Jetson Nano GPU	n/a	49k	59k	19k	87k	n/a	n/a

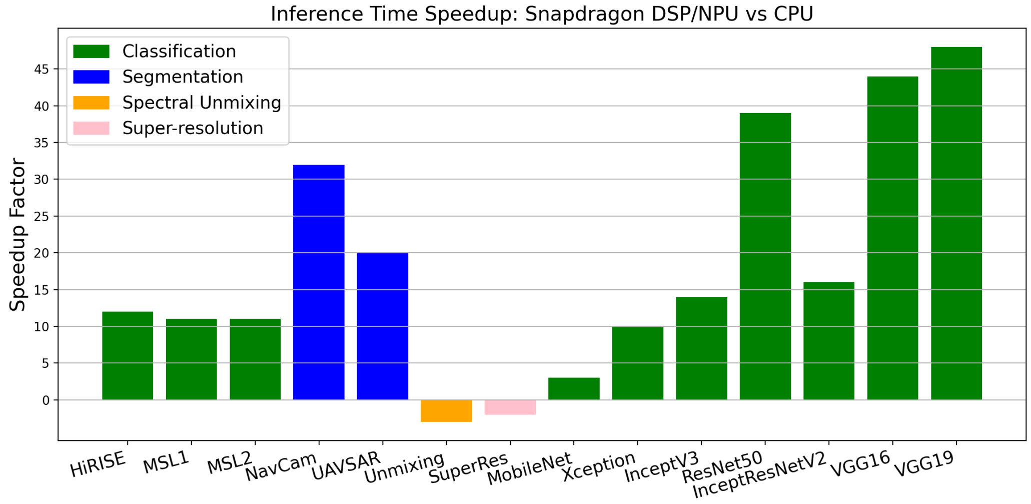


Fig. 10 Inference speedup using the Snapdragon DSP or NPU versus the Snapdragon CPU.

and have similar speedup; the Snapdragon DSP/NPU had greater than 10x speedup over the CPU for all three models. We also benchmarked some standard Keras classification models using the same input data for all models, and found speedups between 3 to 48x, depending on the network structure. While we found that the number of parameters does not predict inference time on the NPU, as shown in Table 10, it does generally correlate with the speedup factor. In Figure 10, we show the inference time improvements using the Snapdragon DSP or NPU over the CPU. The standard Keras models are ordered by number of parameters, on the right side of the figure, from MobileNet to VGG19; we see that the speedup factor increases as the number of parameters increases, except for InceptionResNetV2.

In addition to the classification models, we benchmarked two image segmentation models: Mars NavCam and UAVSAR flood mapping model. The speedup factor of the Snapdragon DSP over the CPU was 32x for NavCam and 20x for the UAVSAR model.

We also benchmarked two single pixel models: a DCDM Spectral Unmixing model and Super-resolution model.

Inference time for these models took *longer* on the DSP/NPU compared to the CPU. This is also shown in Figure 10. The Snapdragon DSP/NPU is not optimized for small single-pixel models such as these.

In Table 11 we show which models were able to run on the Myriad X, the Snapdragon DSP, and the Snapdragon NPU. The SMICES Scikit-learn Classifiers cannot be run on the Myriad X or the Snapdragon DSP or NPU. We were, however, able to port these directly to the Snapdragon CPU using Python-for-Android by Kivy. All other models could be run on the Myriad X except the NavCam DeepLabV3 model, which had incompatible layers. This model was also not able to be run on the Snapdragon NPU. Over half of the standard Keras models could not be run on the Snapdragon DSP, although they could be run on the NPU; the NPU selects the right component for a given task, so it is more flexible than the DSP.

Table 11 Model Summary

Model	Type	Myriad X	Myriad X discr.	DSP	NPU	DSP/NPU discr.
HiRISE	Classifier	yes	0%	yes	yes	1%
MSL1	Classifier	yes	1%	yes	yes	3%
MSL2	Classifier	yes	1%	yes	yes	3%
NavCam	Image Segmentation	no	n/a	yes	no	9%
UAVSAR	Image Segmentation	yes	2%	yes	yes	1%
SMICES	Scikit-learn Classifiers	no	n/a	no	no	n/a
Unmixing	Single Pixel	yes	0 (RMSE)	yes	yes	0 (RMSE)
SuperRes	Single Pixel	yes	0 (tol 1e-1)	yes	yes	0 (tol 1e-1)
MobileNet	Classifier	yes	15%	no	yes	100%
Xception	Classifier	yes	16%	no	yes	11%
InceptV3	Classifier	yes	11%	no	yes	4%
ResNet50	Classifier	yes	8%	yes	yes	4%
InceptResNetV2	Classifier	yes	9%	no	yes	16%
VGG16	Classifier	yes	1%	yes	yes	0%
VGG19	Classifier	yes	1%	yes	yes	1%

In Table 11 we also show the quantization discrepancy for the runs. The Snapdragon DSP and NPU quantization discrepancies are identical, so we capture those results in a single column. The classification discrepancies are reported as a percentage, the Spectral Unmixing is reported at root mean square error (RMSE), and Super-resolution in terms of a tolerance, as discussed in those sections.

The quantization discrepancy for NavCam was 9% for the Snapdragon DSP. Pre-quantization of the model gave an error so we used run-time quantization, which can affect the quantization discrepancy. Many of the standard Keras classification models had quantization discrepancies greater than 5% for both the Snapdragon NPU and Myriad X. These standard classification models have much higher discrepancies in general than our JPL models, but these models were trained to predict many more classes, so this is not surprising. MobileNet had 100% quantization discrepancy for the Snapdragon NPU. This may be due to the structure of MobileNet which gives greater fluctuation ranges and may be more difficult to quantize. For future work, we plan to investigate the Qualcomm efficiency toolkit [46], which may improve network quantization.

All runs gave the exact same inference results no matter how many times they were run, and whether they were run onboard the ISS or on the ground.

IV. Related Work

In this section, we discuss some prior benchmarking work on Myriad and Snapdragon processors, as well as other edge processors that are being explored.

Operating in a space environment introduces power, mass, and radiation tolerance constraints on computing hardware. Previous reviews of appropriate machine learning algorithms and hardware for space applications have been conducted [47] [48] [49].

In the broader literature, there are many image-based deep learning benchmark results for Myriad and Snapdragon processors. Deep learning models ranging from image classification, deblurring, and super resolution for Smartphones [50] to super-resolution for drone landings [51] have been benchmarked on a Snapdragon 835 (prior generation). In addition, the base station for the Mars Ingenuity Helicopter uses the Snapdragon 801 [11].

The Myriad 2 (previous generation of the Myriad X) has been used for a variety of applications. This includes benchmarking in combination with a Raspberry Pi [52] on applications ranging from real time imagery for vehicular edge computing [53] to detection of pathogens with microscopy imagery [54]. In [55], the Myriad 2 was benchmarked in conjunction with a Field Programmable Gate Array (FPGA) on the PASCAL VOC dataset [56]. Skin cancer detection models [57] and wild animal classification models [58] have also been demonstrated. More similar to our applications, the performance of a star identification algorithm on a Myriad 2 has been compared to other vision processors in [59]. In addition, a cloud detection CNN has been run on a Myriad 2 on the Φ-Sat-1 Earth Observing Satellite. This European Space Agency (ESA) mission was the first demonstration of on-board AI CNN inference on a dedicated chip [3][60].

The Myriad X has been used for tasks ranging from Chinese character recognition [61] to license plate recognition [62] to wildfire prediction [63]. Further benchmarking of models to detect and classify anomalous behaviour in telemetry data has also been performed on the Myriad 2 and Myriad X [64].

Due to the limitations in deep space mission on up/down-link bandwidth and onboard compute, there is interest in using COTS (Commercial Off The Shelf) devices to offload processing from traditional space flight hardware [65]. The offloading of this processing allows for increased data processing at source, an improvement in reaction times of space systems, and a significant reduction in the downlink bandwidth required to relay results with ground operations teams [66]. As algorithms used in space applications increase in both complexity and power consumption, the benefits of offloading this processing from the space flight hardware also increases [67].

Prior flights of traditional machine learning software flew on conventional flight processors. The Autonomous

Sciencecraft on Earth Observing One flew Support Vector (SVM) Machine Learning [68], Random Decision Forest, Bayesian Thresholding, and Salience [69] using a Mongoose V processor. The IPEX Cubesat [70] flew Random Decision Forest Machine Learning and Salience on an ARM Cortex processor and a Gumstix co processor.

There has been additional work on AI frameworks for spacecraft, using non-radiation hardened processors. In Reference [71], a Google Coral Edge Tensor Processing Unit (TPU) [72] based co-processor card is developed with a planned AI demonstration on the ISS, along with a Myriad X, and Xilinx Deep Learning Processor (DPU) [73]. In Reference [74], standard deep learning models, including ResNet and MobileNet, are benchmarked on the Xilinx Versal platform, and reference [75] investigates the performance of a space-grade variant of the Versal. In addition, the RISC-V platform has been radiation tested to assess space-readiness [76].

V. Memory Checkers

As shown in Section III, we found no effects of radiation in our benchmarking tests; there were no differences in results between ground versus ISS runs. To explore this further, we run memory tests on both the Myriad X and Snapdragon. Radiation effects on the ISS are expected to be less than on an Earth Orbiting spacecraft due to the ISS acting as radiation shielding. However, these effects are greater than ground-based runs.

A. Myriad X Memory Checker

The memory system on the Myriad X consists of two main sections - 2.5MB CMX (Connection MatriX) memory and 512MB of DDR (Double Data Rate) memory consisting of 2^{27} 32-bit addresses. The addresses range from 0x80000000 to 0x9FFFFFFF. As each address is a 4-byte/32-bit value, the addresses are incremented by 0x4. For example, the first four addresses are 0x80000000, 0x80000004, 0x80000008, 0x8000000C etc.

The purpose of these tests is to determine the health of the DDR Memory on board the Myriad X. Due to the impact of radiation, elements of the DDR can become corrupted - either the actual memory or the memory addressing. Corrupted memory presents itself in the form of a DDR address not correctly maintaining its value. A common form of corruption is that a bit within a memory address becomes "stuck" as either a 0 or a 1, and even if it is written with the opposing value it will not maintain that value and will always be read back as the stuck value. These memory errors can lead to incorrect inference results, and thus it is important to periodically monitor the health of the DDR. Each of the tests executed as part of this memory test suite consists of two stages - a write stage and a read stage. These stages may be repeated and are described below:

- **Write Stage** - This stage consists of writing the entire DDR Memory with fixed values (these values depend on which test is being executed).
- **Read Stage** - This stage consists of reading back the entire contents of the DDR Memory and comparing the values with the expected results.

Three tests are executed as part of the DDR Memory test suite, with each exercising the memory in a slightly different manner. These tests are exercised extensively on ground prior to flight. Errors are injected into the memory during testing to ensure that errors are detected. The following is performed to each bit in DDR memory:

1) **Standard** - This test writes to each address the address itself. For example, 0x80000000 is written as 0x80000000. Once the write stage has completed, the memory range that was written to is then traversed, the value at each address is read back and compared against the expected value.
2) **Flipped Bits** - This test writes to each address the bit-flipped address. For example, 0x80000000 is written as 0x7FFFFFFF. The checking process is the same as in test 1.
3) **Marching** - Multiple read/write cycles are performed for this test. In each cycle, the same value is written to each address, starting at 0x00000000. For each subsequent cycle, an additional bit is written as 1. This process continues until all bits are set. In the next subsequent cycle, the process of flipping the bits back to 0 commences. The checking process is different to the first two tests in that at the start of each round, the value at each address is read back and compared against the expected value, and the value for the next round is written back to the address. A visual representation of the flipping of the bits for a given addresses is shown in Figure 11. A selection of the written values for each round is given below:
   - Round 0 - 0x00000000
   - Round 1 - 0x00000001
   - Round 32 - 0xFFFFFFFF
   - Round 33 - 0x7FFFFFFF
   - Round 64 - 0x00000000

The tests described were developed to ensure full-coverage testing of the DDR memory, with each bit:

- Written as a 0 as part of a set pattern
- Written as a 1 as part of a set pattern
- Written from a 0 to a 1
- Written from a 1 to a 0

Each of the tests was executed for 2 iterations on the ISS. No memory errors were found.

B. Snapdragon Memory Checker

The memory system on the Snapdragon 855 HDK is 6GB LPDDR4x RAM and 128GB USF2.1 Flash. Two memory tests were developed to check both the static and dynamic aspects of the RAM.

The static test operates by allocating three GB of memory and then waiting one full orbit to check the memory bits. This equates to checking the memory every 90 minutes for any errors. The dynamic memory check is implemented by

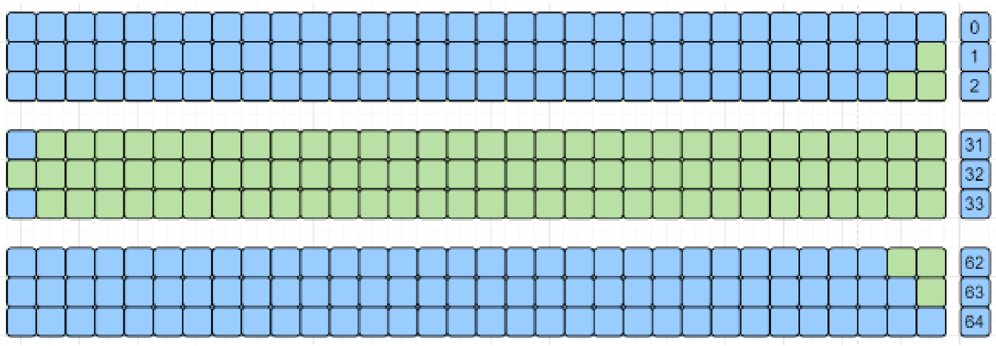


Fig. 11 DDR Marching Memory Test. 0's are represented by blue blocks and ones by green blocks. The round number is shown on the right.

allocating three gigabytes of memory, checking the memory, and then clearing the memory. This process is run as many times as possible. Both of these tests were run over six orbits, or 540 minutes. No memory errors have been found.

VI. Conclusions

We have demonstrated the Myriad X and Snapdragon 855 edge processors for faster and lower power deep learning in space on the ISS. Models benchmarked include image classification, image segmentation, spectral unmixing, and super resolution, and use data from Earth or Mars. The Myriad X VPU and Snapdragon DSP/AIP provide speed improvement over the Snapdragon CPU in all cases except the Spectral Unmixing and Super-resolution single pixel networks. Speedup for the Snapdragon DSP/AIP versus the CPU for non-single pixel JPL models was greater than 10x. Quantization discrepancy was low in all JPL models (less than 5%), except where runtime quantization was used (Mars NavCam model on the Snapdragon DSP).

The standard convolutional neural networks (Mars HiRISE and MSL1 and MSL2) and UNET image segmentation model could be ported to to all Snapdragon processors and the Myriad X. The more complicated DeepLabV3 image segmentation model, however, could not be ported to the Snapdragon NPU or Myriad X due to layer incompatibility.

We also demonstrate Kivy for porting scikit-learn models to the Snapdragon CPU. Using Kivy allows us to run Python code directly on the Snapdragon. This option, however, is not available to the other Snapdragon subsystems or the Myriad X.

Since fine tuning from pretrained networks is a common approach, we benchmarked a set of standard Keras classification models, trained on ImageNet, on the Snapdragon, Myriad X, Jetson Nano, and reference laptop. Run time versus number of network parameters is processor dependent, and is correlated with the speedup factor from the Snapdragon CPU to NPU. Quantization discrepancy was 16% or lower for all models except MobileNet. The discrepancies we find for the models are higher than our JPL models, which is as expected, since the number of classes they are trained on is much higher. The speedups of the Snapdragon DSP/AIP compared with the CPU ranged from 3 - 48x.

We did not observe radiation effects during the running of our models as no discrepancies were found between ground and ISS runs for any of our models.

Future work benchmarking toolkits for the Snapdragon includes comparing the performance of the SMICES multilayer perceptron (MLP) model ported using Kivy, with a DLC model obtained using the Qualcomm Neural Processing SDK. We also plan to investigate Qualcomm's efficiency toolkit [46], which may improve network quantization for our deep learning models.

In addition to comparing results obtained on the ground versus in space, we also ran memory checkers on the Myriad X and Snapdragon 855. We found no errors during our ISS runs. Thus, we have seen no effects from radiation on our processors onboard the ISS. These results, along with the demonstration of a prior version Myriad on an Earth observing satellite [3] [60] and prior version Snapdragon as a base station for Ingenuity on Mars [11], support that these processors may be safe to run in space. In addition, experiments have been performed on the Myriad X and Snapdragon to assess the level of radiation that can be tolerated (see Section II.A).

We have shown fast and accurate inference with these COTS processors and hope this will be a step towards a new era of powerful onboard autonomy with edge processing.

Funding Sources

Portions of this research were carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004).

The JPL element of this work was sponsored by the NASA Science Mission Directorate and the JPL Foundry.

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

Acknowledgments

We would like to thank Damon Russell, Joseph Sauvageau, and Douglas Sheldon for their support via the JPL Foundry Snapdragon initiative. We thank Carrie Knox for her system administration work on HPE's Spaceborne Computer program. We thank Alexandra Holloway from JPL for her work with the RAD750. We would also like to thank the many application providers who also supported the testing of their applications, including: Kiri Wagstaff, Steven Lu, Michael Denbina, Deegan Atha, Michael Swan, and Hiro Ono.

References

[1] "Intel Movidius Vision Processing Units," https://www.intel.com/content/www/us/en/products/details/processors/movidius-vpu/movidius-myriad-x.html, 2022.

[2] “Snapdragon 855 mobile platform,” `https://www.qualcomm.com/snapdragon`, 2022.

[3] Giuffrida, G., Fanucci, L., Meoni, G., Batič, M., Buckley, L., Dunne, A., van Dijk, C., Esposito, M., Hefele, J., Vercruyssen, N., Furano, G., Pastena, M., and Aschbacher, J., “The Φ-Sat-1 Mission: The First On-Board Deep Neural Network Demonstrator for Satellite Earth Observation,” *Institute of Electrical and Electronics Engineers (IEEE) Transactions on Geoscience and Remote Sensing*, Vol. 60, 2021, pp. 5517414–5517414. https://doi.org/10.1109/TGRS.2021.3125567.

[4] “HPE Spaceborne Computer,” https://www.hpe.com/us/en/compute/hpc/supercomputing/spaceborne.html, 2022.

[5] Benton, E. R., and Benton, E., “Space radiation dosimetry in low-Earth orbit and beyond,” *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, Vol. 184, No. 1-2, 2001, pp. 255–294. https://doi.org/10.1016/S0168-583X(01)00748-0.

[6] Akopova, A., Manaseryan, M., Melkonyan, A., Tatikyan, S. S., and Potapov, Y., “Radiation measurement on the International Space Station,” *Radiation Measurements*, Vol. 39, No. 2, 2005, pp. 225–228. https://doi.org/10.1016/j.radmeas.2004.06.013.

[7] “Qualcomm Neural Processing SDK for AI,” https://developer.qualcomm.com/software/qualcomm-neural-processing-sdk, 2022.

[8] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E., “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, Vol. 12, 2011, pp. 2825–2830. URL https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf.

[9] “Python-for-Android by Kivy,” https://kivy.org/doc/stable/guide/packaging-android.html, 2022.

[10] “OpenVINO,” https://docs.openvino.ai/latest/index.html, 2021.

[11] “Journey to Mars: How our collaboration with Jet Propulsion Laboratory fostered innovation,” https://www.qualcomm.com/news/onq/2021/03/17/journey-mars-how-our-collaboration-jet-propulsion-laboratory-fostered-innovation, 2022.

[12] Sheldon, D., Gagne, J., Daniel, A., Vartanian, S., Riley, R., Kallal, E., and Allen, G., “Radiation Effects Characterization and System Architecture Options for the 7nm Snapdragon SA8155P Automotive Grade System on Chip (SoC),” *Radiation and its Effects on Components and Systems Conference*, 2022, pp. 1–16. URL https://ntrs.nasa.gov/citations/20230005645.

[13] “Ubotica CogniSat-TK,” https://ubotica.com, 2022.

[14] Buckley, L., Dunne, A., Furano, G., and Tali, M., “Radiation Test and in Orbit Performance of MpSoC AI Accelerator,” *2022 Institute of Electrical and Electronics Engineers (IEEE) Aerospace Conference (AERO)*, 2022, pp. 1–9. https://doi.org/10.1109/AERO53065.2022.9843440.

[15] “NVIDIA Jetson Nano Developer,” https://developer.nvidia.com/embedded/jetson-nano-developer-kit, 2022.

[16] “AI with Jetson Processors,” https://www.nvidia.com/en-us/autonomous-machines/, 2022.

[17] “Rad750,” https://en.wikipedia.org/wiki/RAD750, 2022.

[18] Developers, “TensorFlow,” https://doi.org/10.5281/zenodo.6574269, May 2022. https://doi.org/10.5281/zenodo.6574269.

[19] “NVIDIA T4,” https://www.nvidia.com/en-us/data-center/tesla-t4/, 2022.

[20] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T., “Caffe: Convolutional Architecture for Fast Feature Embedding,” *Proceedings of the 22nd Association for Computing Machinery (ACM) International Conference on Multimedia*, Association for Computing Machinery, New York, NY, USA, 2014, p. 675–678. https://doi.org/10.1145/2647868.2654889.

[21] “Qualcomm SNPE Software Development Kit Versions,” https://developer.qualcomm.com/sites/default/files/docs/snpe/revision_history.html, 2022.

[22] Chollet, and Team, “Keras Applications,” https://keras.io/api/applications/, 2021.

[23] Dunkel, E., Swope, J., Towfic, Z., Chien, S., Russell, D., Sauvageau, J., Sheldon, D., Romero-Cañas, J., Espinosa-Aranda, J., Buckley, L., Hervas-Martin, E., Fernandez, M., and Knox, C., “Benchmarking Deep Learning Inference of Remote Sensing Imagery on the Qualcomm Snapdragon and Intel Movidius Myriad X Processors Onboard the International Space Station,” *Institute of Electrical and Electronics Engineers (IEEE) International Geoscience and Remote Sensing Symposium (IGARSS)*, 2022, pp. 1–4. https://doi.org/10.1109/IGARSS46834.2022.9884906.

[24] Dunkel, E., Swope, J., Candela, A., West, L., Chien, S., Buckley, L., Romero-Canas, J., Espinosa-Aranda, J. L., Hervas-Martin, E., Towfic, Z., Russell, D., Sauvageau, J., Sheldon, D., Fernandez, M., and Knox, C., “Testing Mars Rover, Spectral Unmixing, and Ship Detection Neural Networks, and Memory Checkers on Embedded Systems Onboard the ISS,” *Proceedings of 16th Symposium on Advanced Space Technologies in Robotics and Automation, European Space Agency*, 2022, pp. 1–8. URL https://ai.jpl.nasa.gov/public/documents/papers/Dunkel-DL-ISS-ASTRA-2022.pdf.

[25] Wagstaff, K., Lu, S., Dunkel, E., Grimes, K., Zhao, B., Cai, J., Cole, S. B., Doran, G., Francis, R., Lee, J., and Mandrake, L., “Mars Image Content Classification: Three Years of NASA Deployment and Recent Advances,” *Proceedings of the Association of Artificial Intelligence (AAAI) Conference on Artificial Intelligence*, Vol. 35, 2021, pp. 15204–15213. https://doi.org/https://doi.org/10.1609/aaai.v35i17.17784.

[26] Krizhevsky, A., Sutskever, I., and Hinton, G. E., “Imagenet Classification with Deep Convolutional Neural Networks,” *Advances in Neural Information Processing Systems*, Vol. 25, 2012. URL https://papers.nips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[27] “Snapdragon Profiler,” https://developer.qualcomm.com/software/snapdragon-profiler, 2022.

[28] Wagstaff, K., Lu, Y., Stanboli, A., Grimes, K., Gowda, T., and Padams, J., “Deep Mars: CNN Classification of Mars Imagery for the PDS Imaging Atlas,” *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI) Conference on Artificial Intelligence*, Vol. 32, 2018, pp. 7867–7872. https://doi.org/10.1609/aaai.v32i1.11404.

[29] Atha, D., Swan, R. M., Didier, A., Hasnain, Z., and Ono, M., "Multi-mission Terrain Classifier for Safe Rover Navigation and Automated Science," *2022 Institute of Electrical and Electronics Engineers (IEEE) Aerospace Conference (AERO)*, IEEE, 2022, pp. 1–13. https://doi.org/10.1109/AERO53065.2022.9843615.

[30] Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H., "Rethinking Atrous Convolution for Semantic Image Segmentation," *arXiv:1706.05587*, 2017. https://doi.org/10.48550/arXiv.1706.05587.

[31] Towfic, Z., Ogbe, D., Sauvageau, J., Sheldon, D., Jongeling, A., Chien, S., Mirza, F., Dunkel, E., Swope, J., Cretu, V., and Ogut, M., "Benchmarking and Testing of Qualcomm Snapdragon System-on-Chip for JPL Space Applications and Missions," *Institute of Electrical and Electronics Engineers (IEEE) Aerospace Conference (AERO)*, IEEE, 2022, pp. 1–12. https://doi.org/10.1109/AERO53065.2022.9843518.

[32] Denbina, M., Towfic, Z. J., Thill, M., Bue, B., Kasraee, N., Peacock, A., and Lou, Y., "Flood Mapping Using UAVSAR and Convolutional Neural Networks," *2020 Institute of Electrical and Electronics Engineers (IEEE) International Geoscience and Remote Sensing Symposium (IGARSS)*, IEEE, 2020, pp. 3247–3250. https://doi.org/10.1109/IGARSS39084.2020.9324379.

[33] Ronneberger, O., Fischer, P., and Brox, T., "U-net: Convolutional Networks for Biomedical Image Segmentation," *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2015, pp. 234–241. https://doi.org/10.1007/978-3-319-24574-4_28.

[34] Swope, J., Chien, S., Bosch-Lluis, X., Yue, Q., Tavallali, P., Ogut, M., Ramos, I., Kangaslahti, P., Deal, W., and Cooke, C., "Using Intelligent Targeting to increase the science return of a Smart Ice Storm Hunting Radar," *International Workshop on Planing & Scheduling for Space (IWPSS)*, 2021. URL https://ai.jpl.nasa.gov/public/documents/papers/Swope-SMICES-targeting-IWPSS-2021.pdf.

[35] Bosch-Lluis, X., Chien, S., Q.Yue, Swope, J., P., P. T., Ogut, M., Ramos, I., Kangaslahti, P., Deal, W., and Cooke, C., "Developing Radiometer and Radar Synergies using Machine Learning," *International Geoscience and Remote Sensing Symposium (IGARSS)*, 2021.

[36] "Advanced Information Systems Technology 2019 awards. Retrieved 20 December 2021," https://esto.nasa.gov/files/solicitations/IIP_19/ROSES2019_IIP_A49_awards.html#deal, December 2021.

[37] Chien, S., Swope, J., Yue, Q., Lluis-Bosch, J., and Deal, W., "Using a Digital Twin Weather Research and Forecasting (WRF) Model for Machine Learning of Deep Convective Ice Storms," *Proceedings of the Fall Meeting of the American Geophysical Union*, American Geophysical Union, Washington, DC, USA, 2021, p. 1. URL https://agu.confex.com/agu/fm21/meetingapp.cgi/Paper/804752.

[38] Candela Garza, A., "Bayesian Models for Science-Driven Robotic Exploration," Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, September 2021. URL https://www.ri.cmu.edu/app/uploads/2021/09/albertoc_phd_ri_2021.pdf.

[39] Hochberg, E. J., and Gierach, M. M., “Missing the Reef for the Corals: Unexpected Trends Between Coral Reef Condition and the Environment at the Ecosystem Scale,” *Frontiers in Marine Science*, Vol. 8, 2021. https://doi.org/10.3389/fmars.2021.727038, URL https://www.frontiersin.org/article/10.3389/fmars.2021.727038.

[40] Thompson, D. R., Hochberg, E. J., Asner, G. P., Green, R. O., Knapp, D. E., Gao, B.-C., Garcia, R., Gierach, M., Lee, Z., Maritorena, S., and Fick, R., “Airborne mapping of benthic reflectance spectra with Bayesian linear mixtures,” *Remote Sensing of Environment*, Vol. 200, 2017, pp. 18–30. https://doi.org/https://doi.org/10.1016/j.rse.2017.07.030, URL https://www.sciencedirect.com/science/article/pii/S0034425717303449.

[41] Candela, A., Thompson, D. R., Wettergreen, D., Cawse-Nicholson, K., Geier, S., Eastwood, M. L., and Green, R. O., “Probabilistic Super Resolution for Mineral Spectroscopy,” *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI) Conference on Artificial Intelligence*, Vol. 34, 2020, pp. 13241–13247. https://doi.org/10.1609/aaai.v34i08.7030.

[42] Swayze, G. A., Clark, R. N., Goetz, A. F., Livo, K. E., Breit, G. N., Kruse, F. A., Sutley, S. J., Snee, L. W., Lowers, H. A., Post, J. L., Stoffregen, R. E., and Ashley, R. P., “Mapping Advanced Argillic Alteration at Cuprite, Nevada, Using Imaging Spectroscopy,” *Economic Geology*, Vol. 109, No. 5, 2014, pp. 1179–1221. https://doi.org/10.2113/econgeo.109.5.1179.

[43] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L., “Imagenet: A Large-Scale Hierarchical Image Database,” *2009 Institute of Electrical and Electronics Engineers (IEEE) Conference on Computer Vision and Pattern Recognition*, IEEE, 2009, pp. 248–255. URL https://image-net.org/static_files/papers/imagenet_cvpr09.pdf.

[44] “Imagenette,” https://github.com/fastai/imagenette, 2021.

[45] Yun, S., and Wong, A., “Do All MobileNets Quantize Poorly? Gaining Insights into the Effect of Quantization on Depthwise Separable Convolutional Networks Through the Eyes of Multi-scale Distributional Dynamics,” *Proceedings of the Institute of Electrical and Electronics Engineers (IEEE)/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 2447–2456. URL https://openaccess.thecvf.com/content/CVPR2021W/MAI/papers/Yun_Do_All_MobileNets_Quantize_Poorly_Gaining_Insights_Into_the_Effect_CVPRW_2021_paper.pdf.

[46] “AI Model Efficiency Toolkit,” https://developer.qualcomm.com/software/ai-model-efficiency-toolkit, 2022.

[47] Murphy, J., Ward, J. E., and Mac Namee, B., “Machine Learning in Space: A Review of Machine Learning Algorithms and Hardware for Space Applications.” *Artificial Intelligence and Computer Science (AICS)*, 2021, pp. 72–83. URL http://star.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-3105/paper21.pdf.

[48] Bruhn, F. C., Tsog, N., Kunkel, F., Flordal, O., and Troxel, I., “Enabling radiation tolerant heterogeneous GPU-based onboard data processing in space,” *CEAS Space Journal*, Vol. 12, No. 4, 2020, pp. 551–564. https://doi.org/10.1007/s12567-020-00321-9.

[49] Furano, G., Meoni, G., Dunne, A., Moloney, D., Ferlet-Cavrois, V., Tavoularis, A., Byrne, J., Buckley, L., Psarakis, M., Voss, K.-O., and Fanucci, L., “Towards the Use of Artificial Intelligence on the Edge in Space Systems: Challenges and Opportunities,” *Institute of Electrical and Electronics Engineers (IEEE) Aerospace and Electronic Systems Magazine*, Vol. 35, No. 12, 2020, pp. 44–56. https://doi.org/10.1109/MAES.2020.3008468.

[50] Ignatov, A., Timofte, R., Kulik, A., Yang, S., Wang, K., Baum, F., Wu, M., Xu, L., and Van Gool, L., “AI Benchmark: All About Deep Learning on Smartphones in 2019,” *2019 Institute of Electrical and Electronics Engineers (IEEE) International Conference on Computer Vision Workshops (ICCV Workshops)*, 2019, pp. 3617–3635. https://doi.org/10.1109/ICCVW.2019.00447.

[51] Truong, N. Q., Nguyen, P. H., Nam, S. H., and Park, K. R., “Deep Learning-Based Super-Resolution Reconstruction and Marker Detection for Drone Landing,” *Institute of Electrical and Electronics Engineers (IEEE) Access*, Vol. 7, 2019, pp. 61639–61655. https://doi.org/10.1109/ACCESS.2019.2915944.

[52] “Rasbperry Pi 3 Model B,” https://www.raspberrypi.com/products/raspberry-pi-3-model-b/, 2022.

[53] Hochstetler, J., Padidela, R., Chen, Q., Yang, Q., and Fu, S., “Embedded Deep Learning for Vehicular Edge Computing,” *2018 Institute of Electrical and Electronics Engineers/Association for Computing Machinery (IEEE/ACM) Symposium on Edge Computing (SEC)*, 2018, pp. 341–343. https://doi.org/10.1109/SEC.2018.00038.

[54] Sethi, K., Parmar, V., and Suri, M., “Low-Power Hardware-Based Deep-Learning Diagnostics Support Case Study,” *2018 Institute of Electrical and Electronics Engineers (IEEE) Biomedical Circuits and Systems Conference (BioCAS)*, 2018, pp. 1–4. https://doi.org/10.1109/BIOCAS.2018.8584697.

[55] Çambay, V. Y., Uçar, A., and Arserim, M. A., “Object Detection on FPGAs and GPUs by Using Accelerated Deep Learning,” *2019 International Artificial Intelligence and Data Processing Symposium (IDAP)*, 2019, pp. 1–5. https://doi.org/10.1109/IDAP.2019.8875870.

[56] Hoiem, D., Divvala, S. K., and Hays, J. H., “Pascal VOC 2008 challenge,” *World Literature Today*, Vol. 24, 2009.

[57] Sahu, P., Yu, D., and Qin, H., “Apply Lightweight Deep Learning on Internet Of Things for Low-Cost and Easy-to-Access Skin Cancer Detection,” *Medical Imaging 2018: Imaging Informatics for Healthcare, Research, and Applications*, Vol. 10579, International Society for Optics and Photonics (SPIE), 2018, pp. 254–262. https://doi.org/10.1117/12.2293350.

[58] Meena, S. D., and Agilandeeswari, L., “Adaboost Cascade Classifier for Classification and Identification of Wild Animals using Movidius Neural Compute Stick,” *International Journal of Engineering and Advanced Technology (IJEAT)*, Vol. 9, No. 13, 2019, pp. 495–499. https://doi.org/10.35940/ijeat.A1089.1291S319.

[59] Agarwal, S., Hervas-Martin, E., Byrne, J., Dunne, A., Luis Espinosa-Aranda, J., and Rijlaarsdam, D., “An Evaluation of Low-Cost Vision Processors for Efficient Star Identification,” *Sensors*, Vol. 20, No. 21, 2020, p. 6250. https://doi.org/10.3390/s20216250.

[60] Giuffrida, G., Diana, L., de Gioia, F., Benelli, G., Meoni, G., Donati, M., and Fanucci, L., “Cloudscout: A Deep Neural Network for On-Board Cloud Detection on Hyperspectral Images,” *Remote Sensing*, Vol. 12, No. 14, 2020, p. 2205. https://doi.org/10.33390/rs12142205.

[61] Kljucaric, L., Johnson, A., and George, A. D., “Architectural Analysis of Deep Learning on Edge Accelerators,” *2020 Institute of Electrical and Electronics Engineers (IEEE) High Performance Extreme Computing Conference (HPEC)*, 2020, pp. 1–7. https://doi.org/10.1109/HPEC43674.2020.9286209.

[62] Tham, M.-L., and Tan, W. K., "IoT Based License Plate Recognition System Using Deep Learning and OpenVINO," *2021 4th International Conference on Sensors, Signal and Image Processing*, 2021, pp. 7–14. https://doi.org/10.1145/3502814.3502816.

[63] Thangavel, K., Spiller, D., Sabatini, R., Amici, S., Sasidharan, S. T., Fayek, H., and Marzocca, P., "Autonomous Satellite Wildfire Detection Using Hyperspectral Imagery and Neural Networks: A Case Study on Australian Wildfire," *Remote Sensing*, Vol. 15, No. 3, 2023, p. 720. https://doi.org/10.3390/rs15030720.

[64] Murphy, J., Ward, J. E., and Namee, B. M., "Developing Machine Learning Models for Space Based Edge AI Platforms," *36th Annual Small Satellite Conference*, 2022. URL https://digitalcommons.usu.edu/smallsat/2022/all2022/79/.

[65] Furano, G., Meoni, G., Dunne, A., Moloney, D., Ferlet-Cavrois, V., Tavoularis, A., Byrne, J., Buckley, L., Psarakis, M., Voss, K.-O., and Fanucci, L., "Towards the Use of Artificial Intelligence on the Edge in Space Systems: Challenges and Opportunities," *IEEE Aerospace and Electronic Systems Magazine*, Vol. 35, 2020, pp. 44–56. https://doi.org/10.1109/MAES.2020.3008468.

[66] Navarro, J., Samuelsson, A., Gingsjö, H., Barendt, J., Dunne, A., Buckley, L., Reisis, D., Kyriakos, A., Papatheofanous, E., Bezaitis, C., Matthijs, P., Ramos, J., and Steenari, D., "High-Performance Compute Board—A Fault-Tolerant Module for On-Board Vision Processing," *European Workshop on On-Board Data Processing (OBDP2021)*, 2021, pp. 1–7. URL https://ubotica.com/wp-content/uploads/2021/08/10.05_OBDP2021_Espana_Navarro.pdf.

[67] Leon, V., Bezaitis, C., Lentaris, G., Soudris, D., Reisis, D., Papatheofanous, E.-A., Kyriakos, A., Dunne, A., Samuelsson, A., and Steenari, D., "FPGA and VPU Co-Processing in Space Applications: Development and Testing with DSP/AI Benchmarks," *2021 28th Institute of Electrical and Electronic Engineers (IEEE) International Conference on Electronics, Circuits, and Systems (ICECS)*, 2021, pp. 1–5. https://doi.org/10.1109/ICECS53924.2021.9665462.

[68] Chien, S., Sherwood, R., Tran, D., Cichy, B., Rabideau, G., Castano, R., Davies, A., Mandl, D., Frye, S., Trout, B., Shulman, S., and Boyer, D., "Using Autonomy Flight Software to Improve Science Return on Earth Observing One," *Journal of Aerospace Computing, Information, and Communication (JACIC)*, Vol. 2, 2005, pp. 196–216. URL https://ai.jpl.nasa.gov/public/documents/papers/chien-JACIC2005-UsingAutonomy.pdf.

[69] Wagstaff, K., Chien, S., Altinok, A., Rebbapragada, U., Thompson, D., Schaffer, S., and Tran, D., "Cloud Filtering and Novelty Detection using Onboard Machine Learning for the EO-1 Spacecraft," *International Symposium on Artificial Intelligence, Robotics, and Automation for Space (ISAIRAS 2018)*, Madrid, Spain, 2018, pp. 1–4. URL https://ai.jpl.nasa.gov/public/papers/wagstaff-ijcai2017-novelty.pdf, also appears at AI in the Oceans and Space Workshop, International Joint Conference on Artificial Intelligence (IJCAI 2017).

[70] Chien, S., Doubleday, J., Thompson, D. R., Wagstaff, K., Bellardo, J., Francis, C., Baumgarten, E., Williams, A., Yee, E., Stanton, E., and Piug-Suari, J., "Onboard Autonomy on the Intelligent Payload Experiment CubeSat Mission," *Journal of Aerospace Information Systems (JAIS)*, Vol. 14, No. 6, 2016, pp. 307–315. https://doi.org/10.2514/1.I010386.

[71] Goodwill, J., Crum, G., MacKinnon, J., Brewer, C., Monaghan, M., Wise, T., and Wilson, C., "NASA SpaceCube Edge TPU

SmallSat Card for Autonomous Operations and Onboard Science-Data Analysis," *35th Annual Small Satellite Conference*, AIAA, 2021, pp. 1–13. URL https://digitalcommons.usu.edu/smallsat/2021/all2021/185/.

[72] "Google Coral," https://coral.ai/products/, 2020.

[73] Xilinx, "Zynq DPU v3.3," Xilinx Corporation, PG338 v3.3, 2021.

[74] Perryman, N., Wilson, C., and George, A., "Evaluation of Xilinx Versal Architecture for Next-Gen Edge Computing in Space," *2023 Institute of Electrical and Electronics Engineers (IEEE) Aerospace Conference*, 2023. https://doi.org/10.1109/AERO55745.2023.10115906.

[75] Petry, M., Gest, P., Koch, A., Ghiglione, M., and Werner, M., "Accelerated Deep-Learning inference on FPGAs in the Space Domain," *Association for Computing Machinery (ACM) International Conference on Computing Frontiers*, 2023. https://doi.org/10.1145/3587135.3592763.

[76] Cannizzaro, M., and George, A., "Evaluation of RISC-V Silicon Under Neutron Radiation," *2023 Institute of Electrical and Electronics Engineers (IEEE) Aerospace Conference*, 2023. https://doi.org/10.1109/AERO55745.2023.10115689.