TESTING MARS ROVER, SPECTRAL UNMIXING, AND SHIP DETECTION NEURAL NETWORKS, AND MEMORY CHECKERS ON EMBEDDED SYSTEMS ONBOARD THE ISS

Emily Dunkel[1], Jason Swope[1], Alberto Candela[1], Lauren West[1], Steve Chien[1], Léonie Buckley[2], Juan Romero-Cañas[2], Jose Luis Espinosa-Aranda[2], Elena Hervas-Martin[2], Zaid Towfic[1], Damon Russell[1], Joseph Sauvageau[1], Douglas Sheldon[1], Mark Fernandez[3], and Carrie Knox[3]

[1]*Jet Propulsion Laboratory* California Institute of Technology, USA*
[2]*Ubotica Technologies, Ireland*
[3]*Hewlett Packard Enterprise*

ABSTRACT

Future space missions can benefit from processing imagery onboard to detect science events, create insights, and respond autonomously. This capability can enable the discovery of new science. One of the challenges to this mission concept is that traditional space flight hardware has limited capabilities and is derived from much older computing in order 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. They offer direct hardware acceleration for deep neural networks, which are state-of-the art in computer vision. We deploy neural network models on these processors hosted by Hewlett Packard Enterprise's Spaceborne Computer-2 onboard the International Space Station (ISS). We benchmark a variety of algorithms on these processors. The models are run multiple times on the ISS to see if any errors develop. In addition, we run a memory checker to detect radiation effects on the embedded processors.

Key words: Deep Learning, Edge Processing, Space Applications, Machine Learning, Artificial Intelligence, COTS embedded processors.

1. INTRODUCTION

Deep space missions have limited contact with ground operations teams, making it hard to account for a dynamic environment. This is due to the limited 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. But 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 promises more powerful onboard autonomy with edge processing.

We benchmark deep learning inference on Movidius Myriad X and Snapdragon processors onboard the ISS. Hosting of these processors is enabled by Spaceborne Computer-2 (SBC-2) by Hewlett Packard Enterprise [4]. Previously, these models have been deployed on the ground. We run inference with these models at various times to see if errors occur, and in addition, run algorithms specifically designed to find memory errors, to get an idea of possible radiation effects.

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.

2. 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 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]. For more information about the Snapdragon 855, please see [5]. Snapdragon processors have been used in vehicles, drones, and even the Mars Ingenuity Helicopter and base station [6].

The Myriad X Vision Processing Unit (VPU) features a Neural Compute Engine, which is a dedicated hardware accelerator for performing neural networks inference, as well as VLIW SIMD cores for accelerating computer vision algorithms. The VPU is programmable using Ubotica's CVAI Toolkit™. Half precision floating point is supported. The previous generation VPU, the Myriad 2, flew on the on the PhiSat-1 satellite, a CubeSat mission from the European Space Agency [3].

We compare results from the Snapdragon and Myriad processors with an NVIDIA Jetson Nano and test laptop, both on the ground. The NVIDIA Jetson Nano features a 128-core NVIDIA Maxwell™ GPU and Quad-core ARM® A57 CPU that operates at 1.43 GHz. The test laptop is a 2019 MacBook Pro, with a 2.4 GHz and 8-Core Intel i9 processor, running Ubuntu 18.04 in a docker container. For future work, we plan to benchmark with two additional more traditional ground testbed processors: Rad750 [7] (used on many prior missions) and Sabertooth [8] (which is being developed for future missions).

3. EXPERIMENTAL SETUP AND ISS DEPLOYMENT

Two Snapdragon 855 development boards (with radios disabled) and two Movidius Myriad X Processors were integrated with the HPE Spaceborne Computer-2, which was launched on February 20th, 2021, as part of the Cygnus NG-15 resupply mission to the ISS. 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 port trained deep learning models to formats that can be run on the Myriad X and/or the Snapdragon, and test locally on these processors. Snapdragon deep learning models are ported using the Qualcomm Neural Processing Software Development Kit [9]. Myriad models are ported using OpenVINO [10] and the Ubotica CVAI Toolkit [11]. See Figures 1 and 2 for pictures of our local hardware.

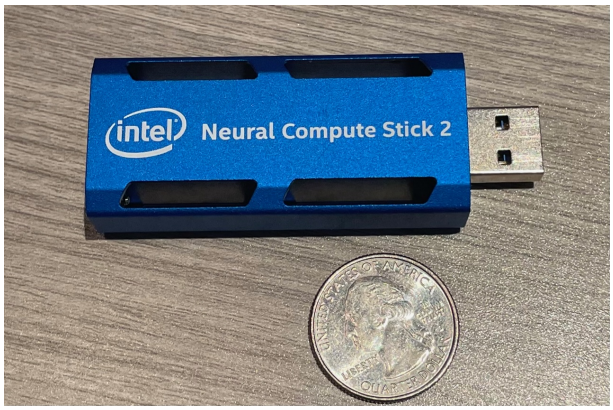

Figure 1. Movidius Myriad X.

Figure 2. Snapdragon 855 Development Board.

Once those models have been verified, JPL runs these models on the HPE Ground Testbed, and then sends HPE a test harness script to run. HPE tests on their Flight Testbed (ground) before deploying on SBC-2 on the ISS. Results are then sent back to JPL for interpretation.

4. PRIOR BENCHMARKING WORK

In [12], we show benchmarks for a set of deep learning models. We show energy and inference time on the Snapdragon and Myriad X for Mars HiRISENET, which is used to classify images collected by the High Resolution Imaging Experiment (HiRISE) instrument onboard the Mars Reconnaissance Orbiter [13]. We found that the low SWaP processors had only small errors (from the quantization), with over 10x speed improvement compared with the Snapdragon CPU. Compared to our test laptop (MacOS 2019, 2.4 GHZ, 8-core), which required 2.3 J (includes monitor and other externals), the Snapdragon CPU required 0.5 J, but the DSP only required 0.016 J, the Snapdragon NPU required 0.014 J, and the Myriad X required 0.032 J.

In addition to Mars HiRISENET, we benchmarked an image segmentation model trained on imagery from the Mars Science Laboratory (MSL) rover's Navigation Cameras [14]. The Snapdragon DSP was not able to use a pre-quantized model, and this lead to high errors (9.3% missed pixels). Also, the model had incompatible layers with the Myriad X. Coming back to Earth, we benchmarked a UAVSAR model trained to detect flooding [15]. Error rates on the Snapdragon DSP/NPU and Myriad X were small, with greater than 6x speed improvement over the test laptop. We also benchmarked a single pixel model for super resolution [16] but found this ran slower on the low SWaP processors, likely due to the net's small size and single-pixel nature.

In addition to our models for specific applications, we benchmarked a set of standard deep learning models for classification [17]. Transfer learning from pre-trained models is often used for model development, so these results may help inform model choice.

In this paper, we present results for additional models more recently tested. We also present results from memory check tests.

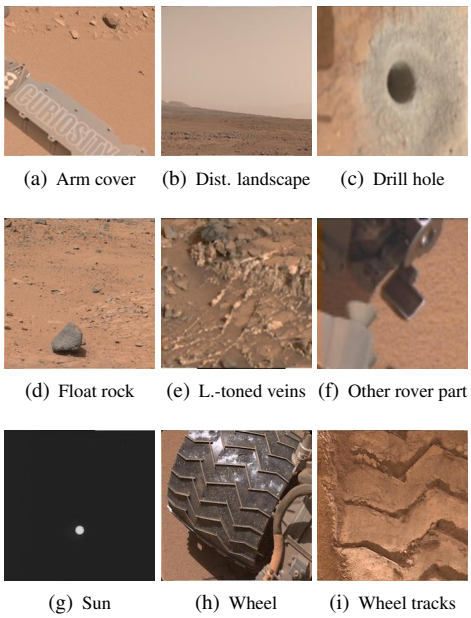

Figure 3. Example imagery from MSL Curiosity Rover's science cameras [13].

5. DEEP LEARNING MODELS AND BENCHMARKS

This paper shows benchmarked results for models trained on Mars and Earth-based imagery. The Mars imagery in this paper is from the MSL rover's science cameras, as opposed to the navigation cameras as in our prior work [12]. For Earth-based models, we look at a model for cloud classification, a model that predicts mixtures of coral, algae, and sand, as well as a segmentation model to detect ships in the ocean.

5.1. Mars MSLNets

The NASA Planetary Data System (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 Mars Science Laboratory (MSL) Curiosity rover's science cameras. Users can interactively search these images for classes of interest using the PDS Image Atlas, which use the predictions from MSLNet (https://pds-imaging.jpl.nasa.gov/search/).

MSLNet is used to classify 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 [13]. If MSLNet1 predicts "other rover parts", the image will be passed through MSLNet2 for finer grained classification of 24 possible classes [18]. See Figure 3 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.

MSLNets were built with transfer learning from AlexNet [19] using Caffe. 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, both of which can lead to a classification discrepancy. Models are quantized using a separate validation dataset and quantization discrepancy errors are reported on a held-out test set.

Benchmarking results are similar for both classifiers, as they have the same model structure. Table 1 shows errors and timing on a test set of 602 images, relative to a Linux run on the test Mac laptop. Inference time is per image. On the test laptop, the time reported is walltime.

Table 2 shows quantization discrepancy errors and timing for MSLNet2, on 1,305 testset image chipouts.

The Snapdragon GPU has 5x speed improvement from the Snapdragon CPU, and the DSP/NPU are 2x faster than the GPU. The Myriad speed is similar to the Snapdragon GPU. Errors were low for all processors.

Snapdragon models have been run for 9 iterations on the

Table 1. Mars MSL1 Classifier Benchmarks

Processor	Errors	Inference Time
MacBook Reference	-	65.4 ms
Snapdragon CPU	0	86.6 ms
Snapdragon GPU	1 (0.2%)	16.2 ms
Snapdragon DSP	15 (2.5%)	7.6 ms
Snapdragon NPU	15 (2.5%)	7.6 ms
Myriad X	3 (0.5%)	16.1 ms
Jetson Nano CPU	2 (0.33%)	1122ms
Jetson Nano GPU	2 (0.33%)	286ms

Table 2. Mars MSL2 Classifier Benchmarks

Processor	Errors	Inference Time
MacBook Reference	-	69.1 ms
Snapdragon CPU	0	81.6 ms
Snapdragon GPU	1 (0.1%)	16.2 ms
Snapdragon DSP	27 (2.1%)	7.6 ms
Snapdragon NPU	27 (2.1%)	7.6 ms
Myriad X	7 (0.5%)	16.1 ms
Jetson Nano CPU	0	1109ms
Jetson Nano GPU	1 (0.1%)	242ms

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

5.2. Single Pixel Cloud Classifier

SMICES is an instrument concept for a "smart" deep convective storm hunting radar [20] [21] [22]. In the SMICES concept, a lookahead radiometer acquires data to detect deep convective ice storms and a radar is used to study detected storms in greater detail. The SMICES machine learning classification application [23] classifies simulated radiometer data into five separate cloud types to identify the location of the deep convective storms. The application runs a random decision forest (RDF), multi-layer perceptron (MLP), support vector machine (SVM), and naïve Bayes Gaussian classifiers 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. Note that the deep learning classifiers here were not ported with the Qualcomm Neural Processing Software Development Kit (SDK), which is what we use for all other models, but instead are ported using the Python-for-Android routine by Kivy [24]. The runtimes for each classifier are listed in Table 3. Future work involves porting with the Qualcomm SDK and benchmarking on the Snapdragon GPU, DSP, and AIP.

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

Table 3. SMICES Classifier Run Times

Classifier	Reference	Snapdragon CPU	Nano
RDF	0.39	0.5 s	0.5 s
MLP	0.31	0.6 s	1 s
SVM	365 s	1316.7 s	2719 s
Bayes	0.06 s	0.3 s	0.3 s

5.3. 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 work 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) [25], which is a probabilistic deep learning model for learning mixtures of classes. It has been used for spectral unmixing and 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 it's 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 [26]. 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 4). 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 [27]. The dataset that was used for the DCDM consists of 12,000 representative PRISM spectra together with their corresponding coral, algae, and sand fractional abundances. The dataset split was performed pseudo-randomly and as follows: 80% train, 10% validation, and 10% test.

Benchmarking results are shown in Table 4. We show performance on the Snapdragon and test laptop, running in batch mode (all pixels passed in at once). Runtime is per-pixel, and MacBook time is wall time. The quan-

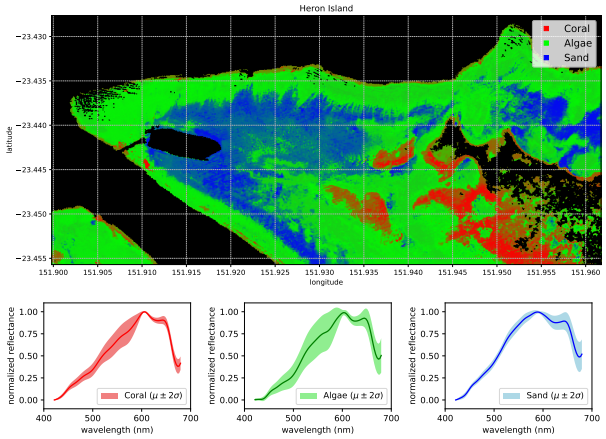

Figure 4. Coral, algae, and sand abundance maps of Heron Island (top) as predicted by the DCDM using PRISM spectra (bottom).

Table 4. DCDM Benchmarks

Processor	RMSE	Inference Time
MacBook Reference	-	0.005 s
Snapdragon CPU	0	0.015 s
Snapdragon GPU	0	0.270 s
Snapdragon DSP	0.05	0.039 s
Snapdragon NPU	0.05	0.039 s

tization discrepancy error is given as root mean square error (RMSE). For this model, the Snapdragon CPU is the faster than the other sub-systems, most likely due to the small size of the model and single-pixel nature. This model has been run once on the ISS, and produced the same results as in the ground-based run.

5.4. Ship Segmentation

The purpose of the Ship Segmentation model is to detect ships in satellite imagery.

The chosen approach uses a Python+PyTorch framework to train a UNet+MobileNetV2 segmentation network. UNet architectures [28] are often used for solving binary semantic segmentation problems. When compared with other encorers, MobileNetV2 [29] has fewer parameters, which makes it easier to train. Furthermore, having a good feature extractor such as MobileNet helps the model to converge much faster during training. The model used pre-trained weights available on the original repository [30].

The dataset chosen is named Airbus Ship Detection dataset, which contains 150,000 JPEG images (768x768 pixels) extracted from Satellite pour l'Observation de la Terre (SPOT) satellite imagery at 1.5 meters resolution [31]. SPOT Image is a worldwide company that focuses on the distribution of products and services using imagery from Earth observation satellites and works

Figure 5. Sample input image and segmented output image for the Ship Segmentation model.

through a network of 30 direct receiving stations, handling images acquired by the SPOT satellites. Images feature tankers, commercial and fishing ships of various shapes and sizes. 70% of the images do not contain ships, and those that do may contain multiple ships, which may differ in size (sometimes significantly) and be located in open sea, at docks, at marinas, etc. 231,723 ship masks were included among 192,556 images. Masks were stored using Run-Length encoding format, allowing segmentation solutions. Images were resized (to 320x320 pixels) for the training stage of this model.

Dataset split was performed pseudo-randomly and the following partitions were created:

- train 90% : 173,302 frames
- val 5% : 9,627 frames
- test 5% : 9,627 frames

The model achieved a dice score of 0.918 over the validation dataset, showing an almost perfect score on large ships. This proved that the image-level information between a ship and the background (sea, shore, beach, etc.) is clearly discernible to this specialised neural network for binarised image segmentation. A sample image and the segmented output is shown in Figure 5.

When deployed on the ISS, a fixed dataset was used with expected results provided. The inference result for each image in the dataset was compared against the expected result. The model has been run on the ISS for two iterations, with 3 times each iteration, and no discrepancies between the achieved and expected inference results were recorded.

For deployment to the Myriad X, the model is 14.8MB in size. A comparison of the performance on the Myriad X, the Myriad 2 (previous generation Myriad) and a CPU are provided in Table 5.

6. MEMORY CHECKERS

We run memory checkers on both the Myriad and Snapdragon to measure the impact of radiation effects on their

Device	Image Size (pixels)			
	160x160	320x320	640x640	768x768
Myriad 2	13.60	4.70	1.26	0.89
Myriad X	23.44	7.56	1.91	1.18
Intel Core i7	59.01	22.64	6.11	4.11

Table 5. Ship Segmentation Performance. Provided results are in Frames per Seconds

memories. The within container Snapdragon and Myriad radiation levels on the ISS are expected to be less than on an Earth Orbiting spacecraft due to the ISS and SBC-2 container acting as radiation shielding.

6.1. Myriad Memory Checker

The memory system on the Myriad X consists of two main sections - 2MB 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 increment 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 the 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 maintaining its correct address. 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. Each of the tests is described below:

1. **Standard** - This test writes to each address the address itself. For example, 0x80000000 is written as 0x80000000. Once the writing 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. In the next subsequent cycle, the process of flipping the bits back to 0 commences. This process continues until all bits are set. 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 6. 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

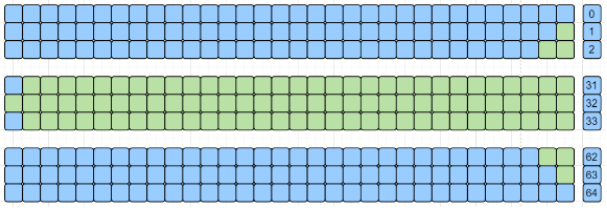

Figure 6. DDR Marching Memory Test. 0's are represented by blue blocks and ones by green blocks. The round number is shown on the right.

No memory errors were found when executed on the ISS.

6.2. 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 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.

7. FUTURE WORK

We continue to benchmark new deep learning applications on the Snapdragon and Myriad X processors within SBC-2 onboard the ISS. Upcoming models include volcano eruption detection and viable road classification for disaster relief. We also plan to benchmark results using Qualcomm's efficiency toolkit [32], which may improve network quantization. In addition, we plan to benchmark on a the RAD750 [7] and Sabertooth [8] conventional flight processors which requires use of deep learning packages. Both processors are currently used for deep-space missions.

8. CONCLUSIONS

We have demonstrated the Myriad X and Snapdragon COTS processors for faster and lower power deep learning in space on the ISS. The Myriad X VPU provides speed improvement over the Snapdragon CPU in all cases, while the Snapdragon DSP/AIP provides speed improvements over the CPU in all cases except the single pixel network.

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.

ACKNOWLEDGMENTS

The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

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.

This work was sponsored by the JPL Foundry and NASA's Earth Science Technology Office (ESTO). We would also like to thank the many application providers who also supported the testing of their applications.

REFERENCES

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

[2] Snapdragon 855 mobile platform. https://www.qualcomm.com/products/snapdragon-855-mobile-platform.

[3] Gianluca Giuffrida, Luca Fanucci, Gabriele Meoni, Matej Batič, Léonie Buckley, Aubrey Dunne, Chris van Dijk, Marco Esposito, John Hefele, Nathan Vercruyssen, et al. The ϕ-sat-1 mission: The first on-board deep neural network demonstrator for satellite earth observation. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2021.

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

[5] Zaid Towfic, Dennis Ogbe, Joseph Sauvageau, Douglas Sheldon, Andre Jongeling, Steve Chien, Faiz Mirza, Emily Dunkel, Jason Swope, Vlad Cretu, and Mehmet Ogut. Benchmarking and testing of qualcomm snapdragon system-on-chip for jpl space applications and missions. *IEEE Aerospace Conf*, 2022.

[6] 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-laboratoryfosterd- innovation.

[7] Rad750. https://en.wikipedia.org/wiki/RAD750.

[8] William Whitaker. Sabertooth: Integrated avionics for small spacecraft missions. In *IEEE Space Computing Conference (SCC)*. IEEE, 2019. URL https://trs.jpl.nasa.gov/bitstream/handle/2014/51550/CL%2319-4553.pdf?sequence=1&isAllowed=y.

[9] Qualcomm Neural Processing SDK for AI. https://developer.qualcomm.com/software/qualcomm-neural-processing-sdk.

[10] OpenVINO. https://docs.openvino.ai/latest/index.html.

[11] Ubotica CVAI. https://ubotica.com/product/specifications/.

[12] Emily Dunkel, Jason Swope, Zaid Towfic, Steve Chien, Damon Russell, Joseph Sauvageau, Douglas Sheldon, Juan Romero-Cañas, Jose Espinosa-Aranda, Léonie Buckley, E. Hervas-Martin, Mark Fernandez, and Carrie Knox. Benchmarking deep learning inference of remote sensing imagery on the qualcomm snapdragon and intel movidius myriad x processors onboard the international space station. In *2022 IEEE International Geoscience and Remote Sensing Symposium*, 2022.

[13] Kiri Wagstaff, Steven Lu, Emily Dunkel, Kevin Grimes, Brandon Zhao, Jesse Cai, Shoshanna B Cole, Gary Doran, Raymond Francis, Jake Lee,

et al. Mars image content classification: Three years of nasa deployment and recent advances. *arXiv preprint arXiv:2102.05011*, 2021.

[14] Deegan Atha, R. Michael Swan, Annie Didier, Zaki Hasnain, and Masahiro Ono. Multi-mission terrain classifier for safe rover navigation and automated science. *IEEE Aerospace Conf*, 2022.

[15] Michael Denbina, Zaid J Towfic, Matthew Thill, Brian Bue, Neda Kasraee, Annemarie Peacock, and Yunling Lou. Flood mapping using uavsar and convolutional neural networks. In *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*, pages 3247–3250. IEEE, 2020.

[16] Alberto Candela, David R Thompson, David Wettergreen, Kerry Cawse-Nicholson, Sven Geier, Michael L Eastwood, and Robert O Green. Probabilistic super resolution for mineral spectroscopy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13241–13247, 2020.

[17] Chollet et al. Keras Applications. `https://keras.io/api/applications/`.

[18] Kiri Wagstaff, You Lu, Alice Stanboli, Kevin Grimes, Thamme Gowda, and Jordan Padams. Deep mars: Cnn classification of mars imagery for the pds imaging atlas. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[20] J. Swope, S. Chien, X. Bosch-Lluis, Q. Yue, P. Tavallali, M. Ogut, I. Ramos, P. Kangaslahti, W. Deal, and C. Cooke. Using intelligent targeting to increase the science return of a smart ice storm hunting radar. *International Workshop on Planing & Scheduling for Space (IWPSS)*, July 2021.

[21] X. Bosch-Lluis, S. Chien, Q.Yue, J. Swope, P. Tavallali P., M. Ogut, I. Ramos, P. Kangaslahti, W. Deal, and C. Cooke. Developing radiometer and radar synergies using machine learning. In *International Geoscience and Remote Sensing Symposium*, July 2021.

[22] 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.

[23] S. Chien, J. Swope, Q. Yue, J. Lluis-Bosch, and W. Deal. Using a digital twin weather research and forecasting (wrf) model for machine learning of deep convective ice storms, 2021. URL `https://agu.confex.com/agu/fm21/meetingapp.cgi/Paper/804752`.

[24] Python-for-Android by Kivy. `https://kivy.org/doc/stable/guide/packaging-android.html`.

[25] Alberto Candela Garza. *Bayesian Models for Science-Driven Robotic Exploration*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, September 2021.

[26] Eric J. Hochberg and Michelle M. Gierach. Missing the reef for the corals: Unexpected trends between coral reef condition and the environment at the ecosystem scale. *Frontiers in Marine Science*, 8, 2021. ISSN 2296-7745. doi: 10.3389/fmars.2021.727038. URL `https://www.frontiersin.org/article/10.3389/fmars.2021.727038`.

[27] David R. Thompson, Eric J. Hochberg, Gregory P. Asner, Robert O. Green, David E. Knapp, Bo-Cai Gao, Rodrigo Garcia, Michelle Gierach, Zhongping Lee, Stephane Maritorena, and Ronald Fick. Airborne mapping of benthic reflectance spectra with bayesian linear mixtures. *Remote Sensing of Environment*, 200: 18–30, 2017. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2017.07.030. URL `https://www.sciencedirect.com/science/article/pii/S0034425717303449`.

[28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[29] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[30] PicsArtHack-binary-segmentation. `https://github.com/gasparian/PicsArtHack-binary-segmentation`.

[31] Airbus Ship Detection Challenge. `https://www.kaggle.com/c/airbus-ship-detection/overview`.

[32] AI Model Efficiency Toolkit. `https://developer.qualcomm.com/software/ai-modelefficiency-toolkit`.