Mission Space Lab Phase 4 report

Team name: Barrande

Chosen theme: Life on Earth

Organisation name: Gymnázium Joachima Barranda Beroun

Country: Czech Republic

1. Introduction

After a bit of brainstorming, we got interested in the Coral module and wanted to explore machine learning in space. We found that identifying Earth topography would be a good use-case for that. We chose Life on Earth for that reason. Our goal was to compare machine learning in space and on Earth. Is the topography classification in space good enough or does it need to happen on Earth later? This finding might be useful for standalone missions with limited hardware, such as in cubesats.

2. Method

We chose a near-IR camera for the experiment, because we expected better data regarding vegetation. We processed images on the Coral module. We also collected data from all other available sensors for a potential secondary mission. We calculated ISS location using the provided library (Figure 1).

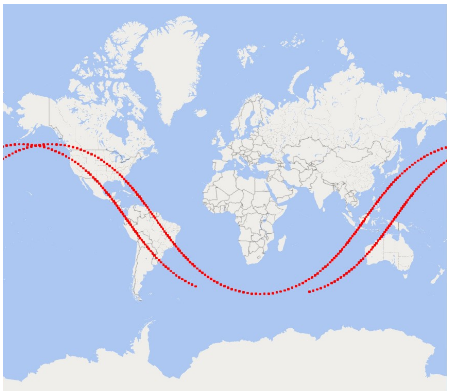

Figure 1: ISS location during the experiment

We decided on identifying 10 different types of terrain that our model would try to recognize: large bodies of water, rivers, clouds, forests, fields/grasslands, deserts, unidentified (land), unidentified (other), islands and mountains. We manually created a training dataset from older Astro Pi photos, by painting coloured masks for 136 different images based on the type of terrain. To increase the size of our training data, each image was rotated by 5°, 71 times, giving us a total of 9792 images in our dataset.

The TensorFlow library was then used to train our model. The model was then converted to TensorFlow lite, to be able to run on the Coral module in real time on the ISS.

3. Experiment results

Our program ran onboard the ISS for three hours without any errors and processed more than 300 images in that time. The processing was happening real-time in under 30 seconds per image.

When we got data back from space we ran different AI model on those images and got results for comparison. In space we had a PAN model with MobileNetV2 as backbone while on Earth we had a custom U-Net model. Since both models were trained on the same dataset we can use statistics to measure how good the models were. The model in space got 75.8% of pixels from the dataset identified correctly and the model we've ran on Earth identified 81.14% pixels correctly, both numbers include the black border around the image itself. You see examples of some images in Figure 4. We've put all images side-to–side into one quick video: https://youtu.be/LtKCgiPmew0.

The AI on Earth did better at recognizing land, ocean and clouds while the AI in space separated the image from the black border and correctly identified clouds. The AI on Earth "over-tried" and created a lot of seemingly random colour spots, mostly in night images (Figure 2).

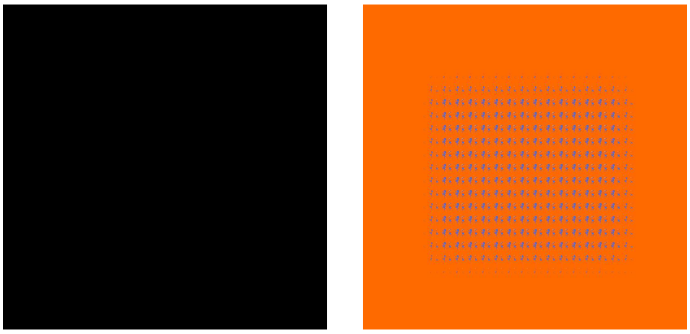

Figure 2: These images show the night side, the left one is original and the right one is the same image as processed by the AI on Earth. The processed one has strange blue patterns in the middle. These patterns do not appear in images processed by the AI in space.

During the flight we also measured and saved values from all the available sensors and after careful examination we didn't find anything anomalous. We hoped to see the ISS propel itself to higher orbit fighting the atmospheric drag, but unfortunately that didn't happen. Another idea was to see how water ecosystems change over time. By comparing images from 1984 (Google Earth) with ours we've found out that the Great Salt Lake has shrunk dramatically (Figure 3).

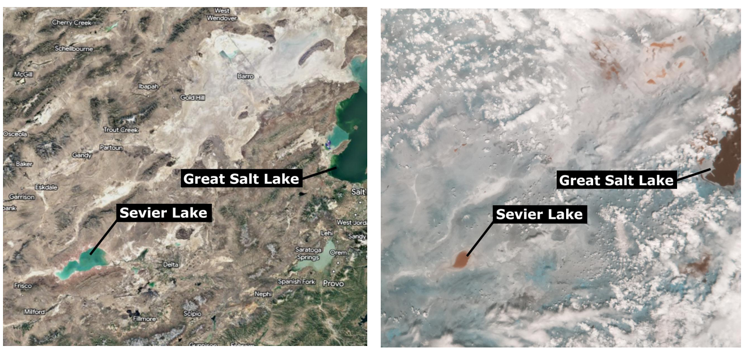

Figure 3: The left image was captured by Landsat satellite in 1984, nearly 40 years ago. The second (right) image captured during our flight shows that the Great Salt Lake and Sevier Lake have lost a lot of water over time.

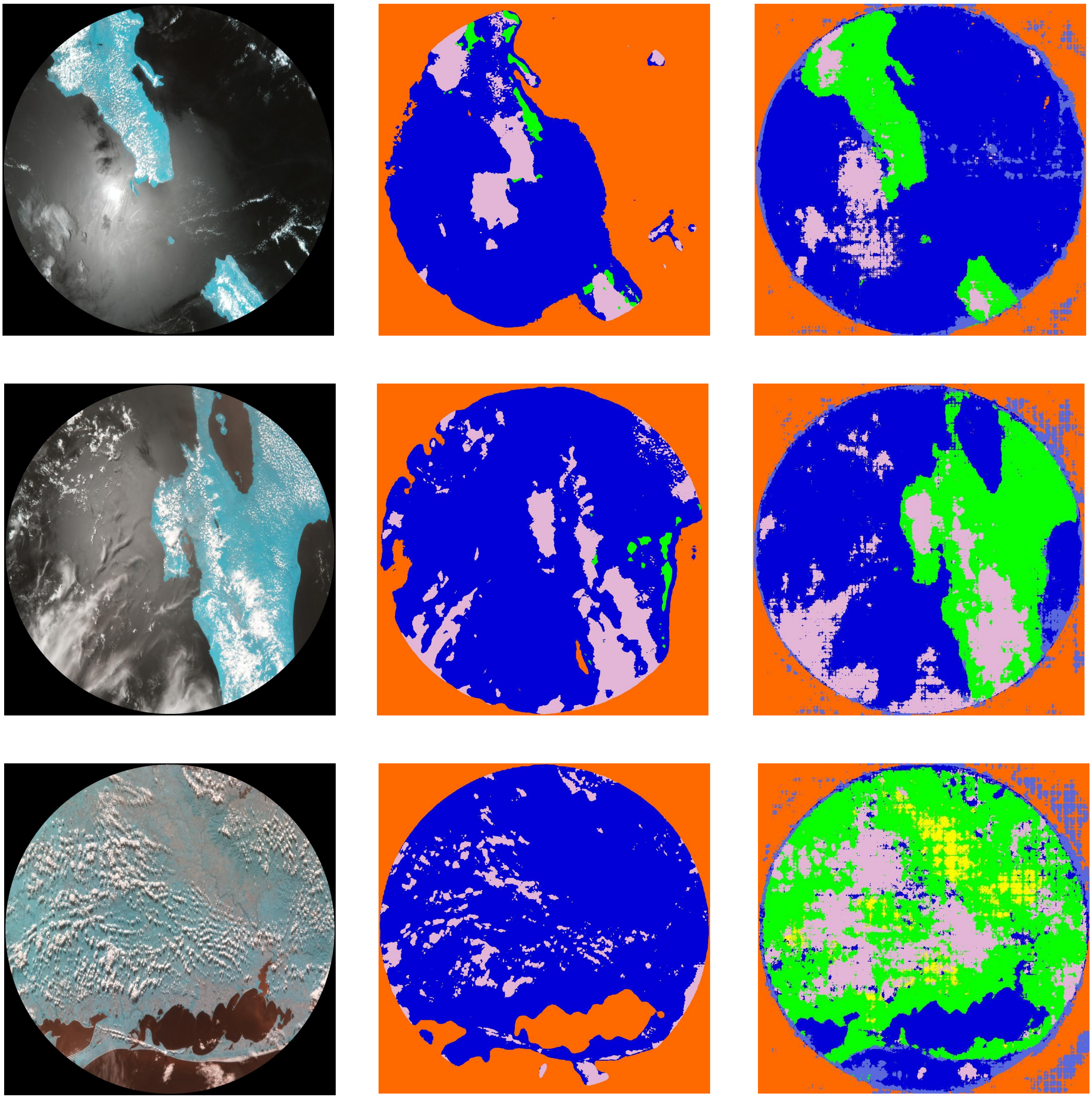

Figure 4: Comparison of images; left - cropped original in NIR spectrum, middle - processed in space, right - processed by better AI on Earth.
First image was taken above Puerto Rico, the second image above Lago Nicaragua (Nicaragua) and the third image above Rio Grande (Brazil).
In the third image AI on Earth incorrectly identified deserts but the AI in space successfully identified only clouds.

4. Learnings

We started at a hackathon organised by planetarium in Prague, where we brainstormed many ideas and created a prototype. Then we used Discord as the main communication tool, GitHub for sharing code. We used Google Spreadsheets to divide our work and set deadlines. We met once more before the Phase 2 deadline for a hackathon in our school.

The biggest challenge was to fit our machine learning model on AstroPi, which had outdated libraries with many issues. When we googled solutions, the answer was usually "just update the library", which we couldn't do. We also contacted AstroPi support without much success. We managed to create a compatible model after many trial-and-error. We had to cut some corners which resulted in lower quality of the model.

It was also challenging to organise work between people with different programming backgrounds. We found out that dividing code into small files, writing comments and clean code helps understanding a lot. If we had a chance to start from the beginning we would create a bigger dataset for our AI and likely improve our results. We wouldn't use the camera with NIR filter again, as it didn't give us any valuable data for our task.

5. Conclusion

Our experiment was successful: we verified that it's possible to run machine learning classification in real-time on a restricted hardware in space, such as in a cubesat. Our processing of the images back on Earth gave better results and for an unexpected reason. We expected issues like lack of processing power or radiation. But at the end it was outdated library versions which we couldn't update on AstroPi.

We learned a lot during the experiment: how to organise a diverse team of students with different programming experience, how to work remotely, how in-person hackathons can speed things up, and how to write clean and separated Python code. And most importantly: our code actually ran in space and we had a lot of fun in the process!