Mid-Term Report

By

Yaseen Lahmami

Advisor

Dr Ian Martin

Project Title

Training an Artificial Intelligence system to detect and classify boulders on images of planetary surfaces and asteroids.

Project Aims and Objectives

The aim of the project is to design a tool to detect and classify boulders on images of planetary surfaces and asteroids, output the size and coordinates of each detected boulder, and use the output to generate an accurate model simulation of rocky terrain ***(figure 1)***. This tool will be developed to target PANGU (Planet and Asteroid Natural Scene Generation Utility). PANGU ***(figure 2)*** is a software that was designed to help simulate and model planetary surfaces. The software helps researchers' and users' model various cases from simulating spacecraft landing, taking images of artificial terrain, and creating planetary models from real world data.

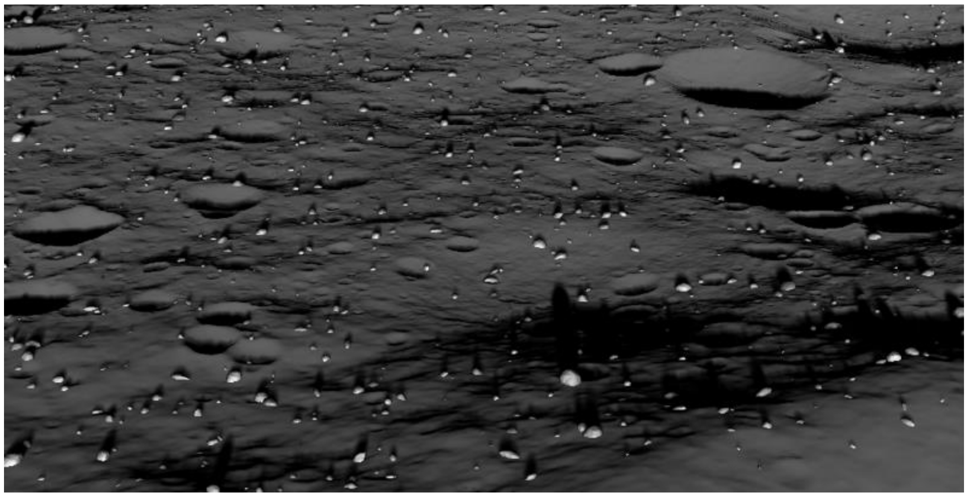
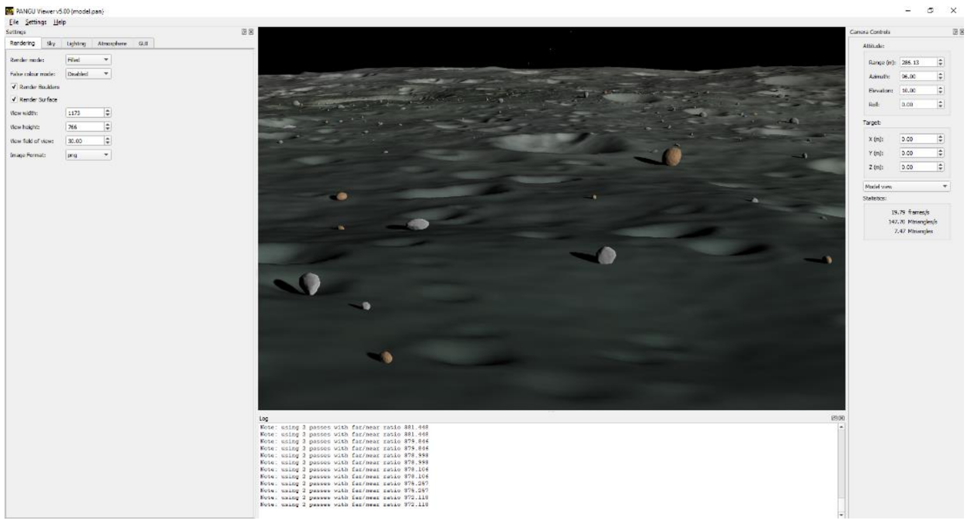


Figure 1– Generated lunar surface model using PANGU. Boulders shown as relatively small rocks scattered on the terrain. The task is to classify, detect and output the results into PANGU as accurately as possible.



Render mode: Filled
False colour mode: Disabled
✓ Render Boulders
✓ Render Surface
View width: 1173
View height: 766
View field of view: 30.00
Image Format: png

Attitude:
Range (m): 286.13
Azimuth: 96.00
Elevation: 10.00
Roll: 0.00

Target:
X (m): 0.00
Y (m): 0.00
Z (m): 0.00

Model view

Statistics:
19.79 frames/s
147.70 Mtriangles/s
7.47 Mtriangles

```
Note: using 3 passes with far/near ratio 881.448
Note: using 3 passes with far/near ratio 881.448
Note: using 3 passes with far/near ratio 879.846
Note: using 3 passes with far/near ratio 879.846
Note: using 3 passes with far/near ratio 878.998
Note: using 3 passes with far/near ratio 878.998
Note: using 3 passes with far/near ratio 878.106
Note: using 3 passes with far/near ratio 878.106
Note: using 3 passes with far/near ratio 876.267
Note: using 3 passes with far/near ratio 876.267
Note: using 3 passes with far/near ratio 872.118
Note: using 3 passes with far/near ratio 872.118
```

Figure 2– PANGU viewer showing a lunar surface model. The boulders shown are generated manually. The task is to automatically generate these boulders to save time and to improve user experience.

My project is targeted to help users save time and increase the accuracy, when creating artificial planetary surfaces from images taken from NASA and other space agencies. This would be beneficial as the user will no longer need to define the size and coordinates for every boulder in the image. PANGU users would find this helpful because the user would save a significant amount of time by not having to manually enter the size and position of each boulder. With a possibility of there being hundreds of boulders per image, it could take the user hours. Automating this process to at least sufficient accuracy will help the user spend more time using the tool for other purposes such as simulations. Accuracy is also important reason to increase the reliability of each simulated model. Manual input may cause miscalculations of properties such as the size and positions of the boulders. This is especially crucial for use cases such as simulating spacecraft landing. Errors in the dimensions of boulders and their positions can likely cause spacecraft to crash land if the simulation was tested in real life. This is just an example of how the accuracy of the model can affect the outcome of real-life projects. I intend to use deep learning and computer vision techniques to help improve the speed and the accuracy of the models generated in PANGU to improve the user experience and reliability of the terrains generated.

Progress and Background Research

Due to the nature of this project, my research consists of mostly deep learning and computer vision. During the early stages of the project, I spent most of my time looking into the basics of computer vision, machine learning and neural networks. Before this project I had some previous knowledge on neural networks and machine learning, but computer vision was a very new subject to me. Computer vision requires an area of machine learning known as deep learning. Between the early stages of semester one till now I have been studying the basics of deep learning, applying my knowledge in small projects and familiarizing myself with different types of networks. There are different classes of neural networks in deep learning. A class of neural network that is commonly used in computer vision applications are known as CNNs (Convolutional Neural Networks). Just like there are many software architectures for different use cases, there are also many different CNN architectures to choose from depending on the problem. This is very important for my project as choosing the right architecture will influence how well my project performs. There are three questions I would need to ask myself before choosing a specific architecture. Is it a classification problem? Is it on object detection problem? Or is it both? First, the program would need to scan the image of a planetary surface and classify whether there is a boulder or not. If there is no boulder then the program continues scanning the image. If there is a boulder then move on to the next stage. In the next stage the program will now try to predict where each boulder is in the image. From this we can tell that I would need an architecture that has a classification part and an object detection part.

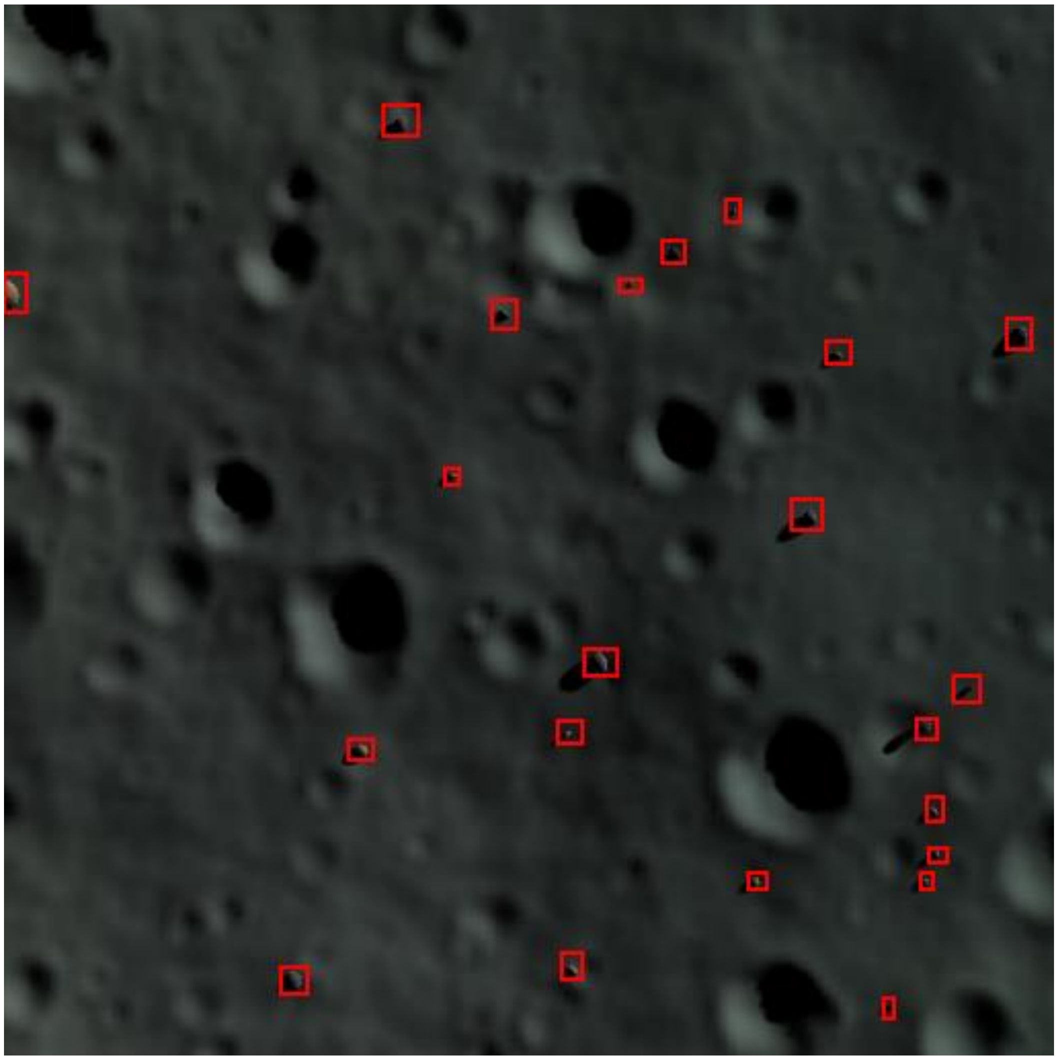


Figure 3– Visual representation of how object detection would work. Each box is a represents detected boulder. This is a lunar surface model generated by PANGU.

There are many architectures to choose from such as R-CNN (Region based CNN), Mask R-CNN (Mask Region based CNN), YOLO (You Only Look Once) and a couple more. These architectures will have a classification module and an object detection module. Before I decide an architecture, object detection is done in different ways. The first way is classic object detection ***(figure 3)*** where if an object is found, a box is drawn around the object in the final output. This is usually applied in self-driving applications where only the detection of objects matter. The drawbacks are that I would not be able to determine the exact shape of the boulder, as the detection would show as rectangular box around the boulder. The second way is image segmentation ***(figure 4).***

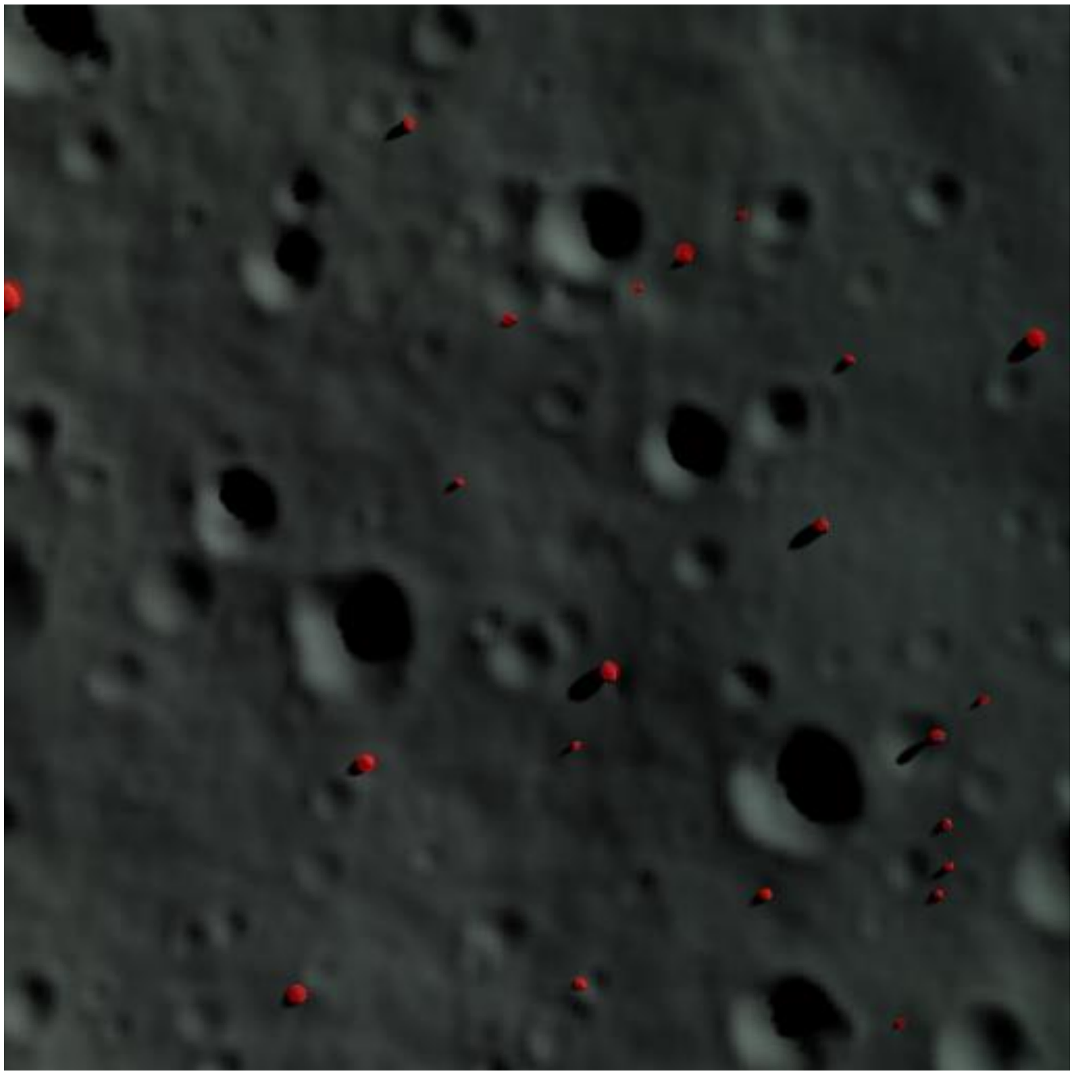


Figure 4– Visual representation of how image segmentation would work. Each dot represents a detected boulder. This is a lunar surface model generated by PANGU.

Image segmentation ***(figure 4)*** is a further extension to object detection where we mark the presence of an object pixel by pixel. This will allow me to get the positions and roughly the size of a boulder without further complications. Some example architectures that are designed for image segmentation are Mask Region based CNN and UNET ***(figure 5)***. Image segmentation is commonly used in medical image diagnosis, autonomous vehicles, and Geo Sensing. As of writing this report, I have yet to choose and architecture however I am close to deciding on one. I am leaning towards image segmentation with either Mask R-CNN or UNET architecture. The reason is due to a similar project to this one that uses image segmentation to detect and classify craters on lunar surfaces. The paper is titled "Lunar Crater Identification via Deep Learning" and proposes a compelling way to highlight regions where a crater is detected. In addition, many applications that observe geo special data use image segmentation to separate different regions. My idea for this project is to use deep learning to classify and detect boulders by using image segmentations to highlight each boulder, which will reveal their size and location. I would like to do a bit more research before I decide to choose image segmentation because I want to make sure that I analysed all the solutions.

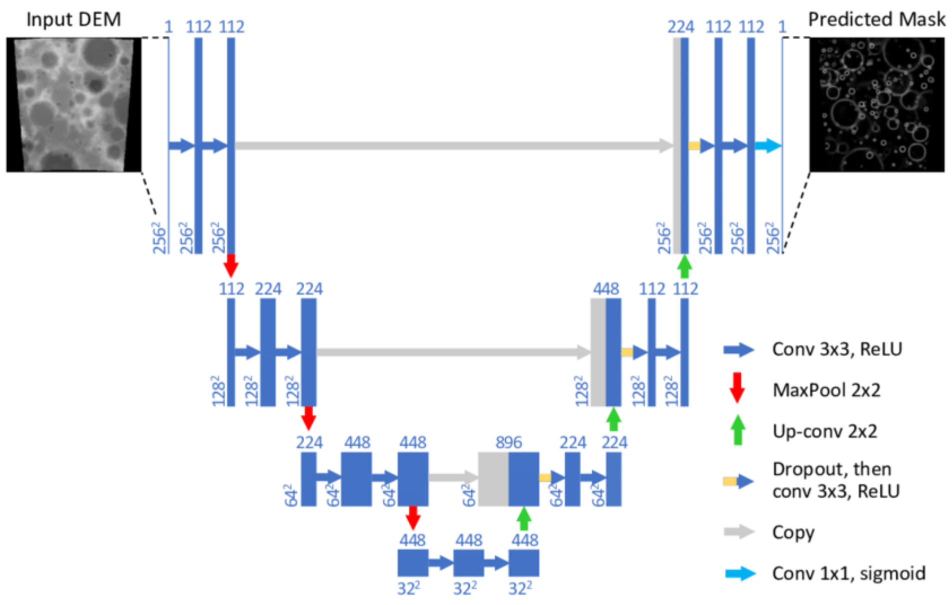


Figure 5- UNET architecture (CNN) that was implemented by the Deep Moon team. The paper "Lunar Crater Identification via Deep Learning" proposed a method of image segmentation (figure 4) to detect and classify craters on the lunar surface.

The areas of my project that I am certain on is the technology stack and the software architecture. The technology stack is as follows:

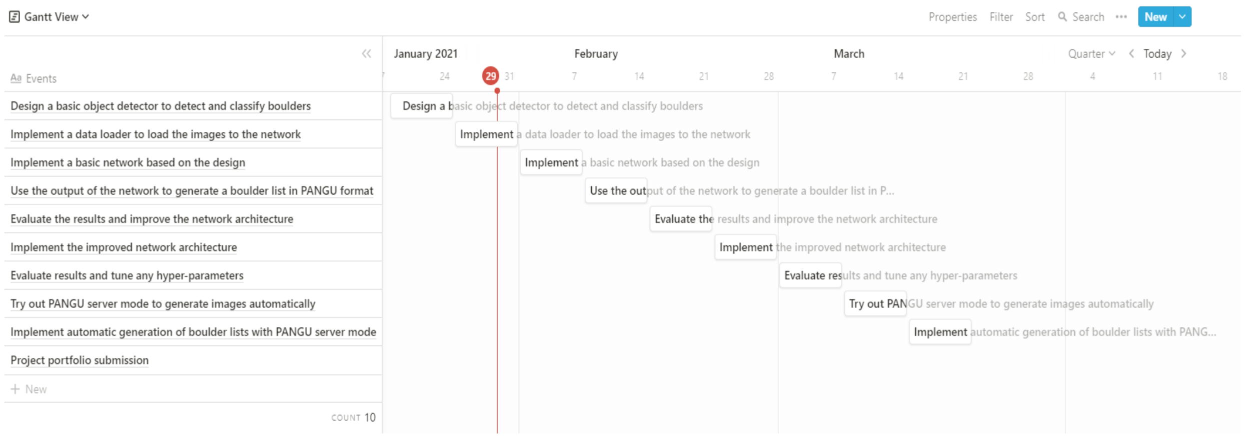
- For deep learning, Pytorch library using Python 3 as it is an ideal choice for beginners due to its fast-learning curve over TensorFlow.
- For image processing and object detection, Open CV library using Python 3. This will allow me to visualise the output by highlighting predicted boulders.
- Matplotlib, a Python library for visualising data.

For the software architecture, PANGU and the deep learning module will be separate for now. The general flow is the deep learning module will act as an input and PANGU will output the result. The deep learning module will input the training images and output predictions of the size and coordinates of each boulder. The output of the deep learning module will act as an input to PANGU. PANGU will use this information to generate a simulated terrain with the predicted boulders.

Apart from time spent on research, the rest of my progress included familiarising myself with the PANGU tool and using the PANGU tool to collect training images. The reason for this is due to there being limited data on real planetary surfaces that is sufficient for this project. I need training images that are detailed enough to show boulders as the sizes of these boulders are significantly smaller than craters. PANGU can also generate a sequence of images from one surface model. This would allow me to gather enough training images – more than what is currently online. Finally, I will be able to compare my results with the surface models boulder list (the ground truth). This will allow me to produce evaluation metrics that will allow me to evaluate where the model is at, and if it needs to be adjusted. To further test my model, I will use the limited real-life images available online to add to the evaluation.

Plans for the remainder of the project

My plans for the remainder of the project will be to use the knowledge and experience gained in computer vision and deep learning to start implementing the project. I have a Gantts chart ***(figure 6)*** detailing the steps that should be completed week by week. By early February I should have a data loader that will pre-process the input images. Between February and March, I will plan to finish the implementation a basic CNN architecture including training, improving, and testing the network. Between the end of February to mid-March, the output of my network will be formatted to be able to use with PANGU. As for the remainder of the project until the deadline, I will look to make any necessary improvements to the model, and I would hope to implement more features. A possible additional feature would be to run PANGU in server mode. This will allow for the tool to directly connect to PANGU, so that the outputs can be loaded onto PANGU automatically.



Design a basic object detector to detect and classify boulders
Implement a data loader to load the images to the network
Implement a basic network based on the design
Use the output of the network to generate a boulder list in PANGU format
Evaluate the results and improve the network architecture
Implement the improved network architecture
Evaluate results and tune any hyper-parameters
Try out PANGU server mode to generate images automatically
Implement automatic generation of boulder lists with PANGU server mode
Project portfolio submission

Figure 6– Gantt's chart showing the important milestones of the project.

Personal reflection

I have had some struggles during the early stages of the project. At first, I struggled to understand the problem I was solving. I knew that this project was trying to solve a computer vision problem, but I misunderstood the difference between a boulder and a crater, which was due to my limited knowledge. This led me to find the wrong images online for testing – I was looking for craters rather than boulders. I also didn't have a clear final picture of what the project was about. I learned more about the project as time went on by playing around with PANGU and learning more about computer vision. Finally, the subject of deep learning and computer vison was quite new to me. Trying to balance the research with my semester modules was quite difficult but not impossible. I managed to balance my time effectively, which helped my progression.

Personally, I feel like the project is going well so far and I am confident that I have understood the problem at hand. I should be on track to complete the project hopefully by late march, so that I can focus the rest of the time to make improvements and adjustments before the deadline at the end of April.

References

Deep Moon

- [Paper] Lunar Crater Identification via Deep Learning - https://arxiv.org/abs/1803.02192
- [GitHub] Lunar Crater Identification via Deep Learning - https://github.com/silburt/DeepMoon

PANGU (Planet and Asteroid Natural Scene Generation Utility)

- [About] PANGU - https://pangu.software/

Deep Learning Libraries

- [Home] Pytorch - https://pytorch.org/
- [Home] TensorFlow - https://www.tensorflow.org/

OpenCV

- [Home] OpenCV - https://opencv.org/

Matplotlib

- [Home] Matplotlib - https://matplotlib.org/