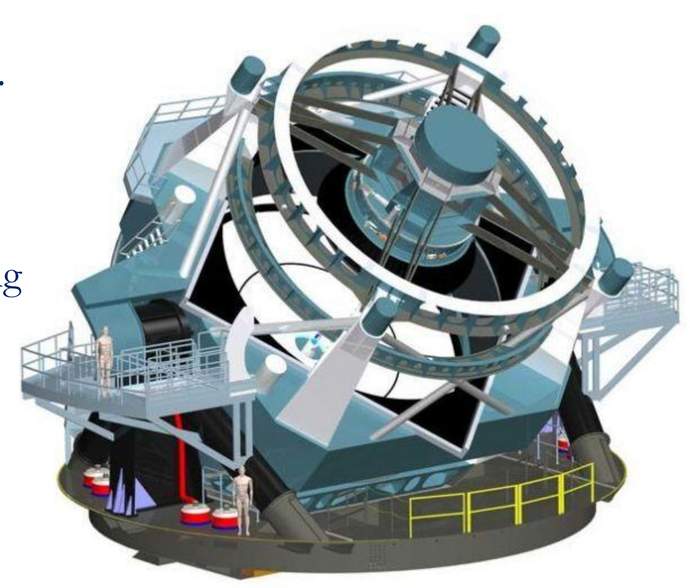
Data and Image Processing using Machine Learning Software for Space Exploration

Sam Hughes, Sumeet Kothare

University of Delaware

Abstract

- Astronomical research will soon take a dramatic turn into the world of big data. The LSST telescope, set to come online in 2022, is aiming to track changes in the night sky on a night to night basis. It will do this by photographing the entire night sky through over one thousand exposures every night taking in over fifteen terabytes of data. Plans for more telescopes like this bring up an important issue, who is going to analyze all that data. Frankly, it's for us.impossible for humans to do by hand which is raising the importance of machine learning and image classification software to do the work.



https://www.popularmechanics.com/space/telescopes/g2961/telescopes-of-the-future/

Introduction

- Crater detection and landform identification software have been developed alongside facial recognition software, making use of similar AI algorithms. Our project aims to recreate some of this software to identify landmarks in images. We think this software could be applied to assist the accuracy of probe landings on foreign planets. Too distant for humans to control in real time, probes must be able to quickly and accurately locate themselves and make necessary adjustments to land at desired sights. One way we think they could do this is identify landmarks on the surface directly below them, compare to previously taken satellite images and determine their exact location and altitude above the surface. With this information, it makes a proper landing more easily attainable.

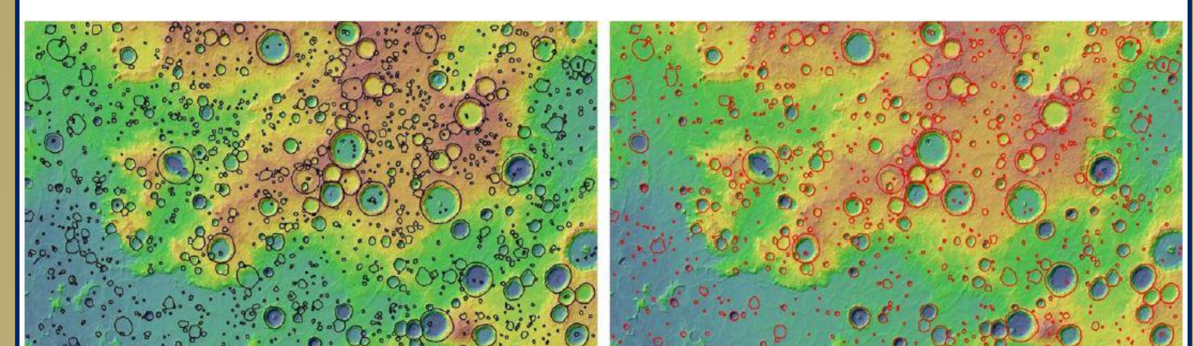
Purpose & Research Question

- Use the Google Machine Learning Engine (MLE) and the TensorFlow Python package, which employ unsupervised and supervised Machine Learning principles, to group similar planetary topographical images.
- Conduct thorough literature search to identify various applications and projects in Data and Image Processing.
- Use the algorithms and data presented in these applications and projects to test the consolidated software program for grouping and categorizing topographical images.

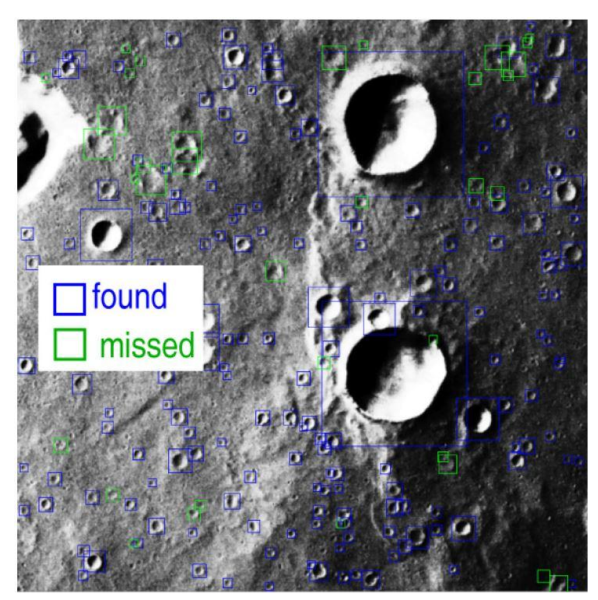
Subjects, Methods & Analysis

Literature Search and Motivation: Detecting Impact Craters in Planetary Images

- Two supervised machine learning algorithms were used.
- Crater Detection Analysis from Digital Elevation Models.



- Crater Detection analysis from panchromatic or grayscale images.

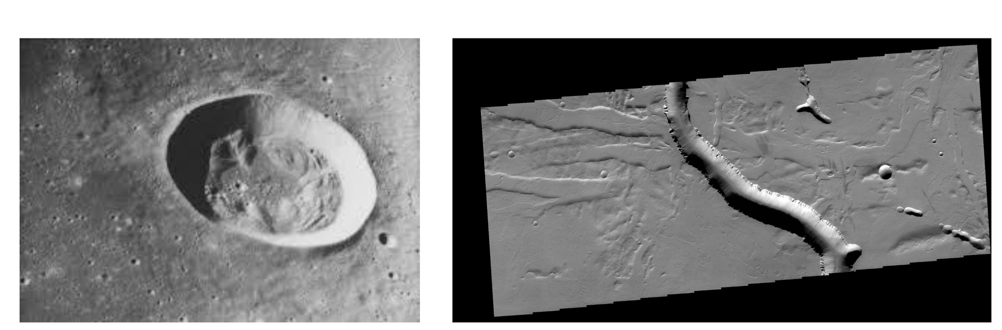


Our Software: Identifying Craters and Non-craters

- The program we aim to develop ourselves will use machine learning to differentiate craters from other surface features in images.
- We will use the Google Colab to platform to write and test our code.
- We based our machine learning algorithm off an example provided by the Google Colab software which takes in data from a CSV file and develops methods of predicting and outcome base on a number of inputs for each test data point

Training Dataset:

- In order to train our machine learning algorithms, we had to acquire a dataset that would uniquely help solve our problem and be compatible with our code.
- We compiled this ourselves by measuring the length and width of features in images of extraterrestrial surfaces.



- Images like these were used in our dataset and the length and width of the main features were measured by hand.

Results

First Try

•Our first try at developing code that could differentiate between craters and other features given the height and width of the feature never proved successful.

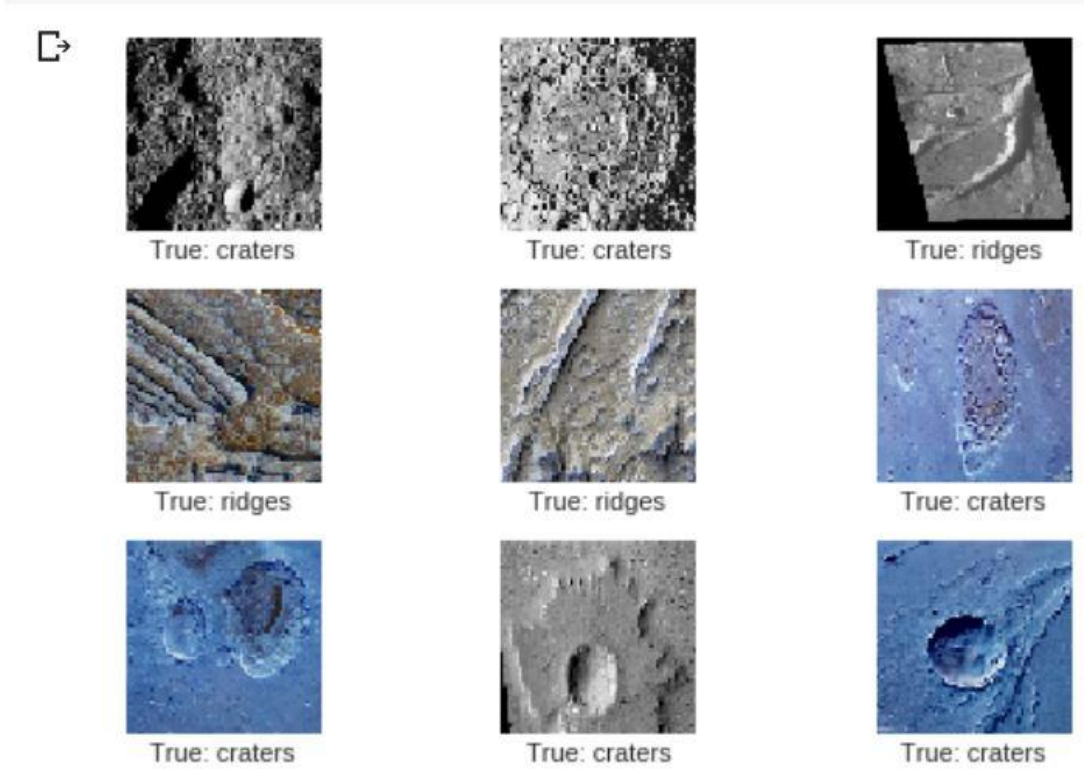
•The code was never able to fully execute because the data set would not import correctly.

Second Try: Convolutional Neural Network

- In a second attempt, we modeled our program off a new example that made use of convolutional neural network.
- In this program we were able to import actual images so that the software could read more details off the images and consider more data points in determining if the feature is a crater or a ridge.
- We also increased the size of our training dataset as well. The images were still picked by hand so we only were able to acquire 52 images.
- This code was able to execute and we able to identify some images correctly.

```
[71] images, cls_true  = data.train.images, data.train.cls

     # Plot the images and labels using our helper-function above.
     plot_images(images=images, cls_true=cls_true)
```



- Using a testing dataset containing 10 images, our initial trained model was only able to produce a 38.5% success rate.
- We attribute this low success rate to having a too small of a training data set.

Conclusions

- Conclusively, the final result of the model of 38.5% accuracy in predicting the correct feature, crater or a ridge, is mediocre.
- This accuracy is heavily dependent on sampling bias and the kind of training data set used to train the model. This project used 52 images to train the mode and then use 10 images to test it.
- The accuracy may be improved by selecting more uniform, consistent, cleaner, and numerous images of topographical features. Clearly, a well-trained model will require a large dataset.
- Furthermore, several images sourced for the project from the internet and from NASA were not clean pictures with always distinguishable or exclusive topographical features. Hence, such training sets may be created in the future to supplement minimally bias-hampered treatment.

Directions for Future Research

- We think this kind of technology has many potential future applications. New astronomical images are being taken at such high rates that photo classification software is going to be a new keystone in astronomical research.
- Our project specifically could be made much more useful by integrating an image fetching program that could pool images from the internet to create larger training sets.