CS6886 Project: To train a model on synthetically generated data and test it on AI4Mars.

Name: Varun Sunderarajan
Roll Number: NS25Z318

This project is broadly divided into two parts:

- Setting up Unity script to make and synthetically generate scenes and their respective labels.
- Train a UNet model on synthetic data generated using Unity and test its performance on the AI4Mars dataset.

Unity scenes

The code simulates a 3D terrain and takes a side view picture with labels including *soil, sand, rock, big-rock*. In the label *Sky and anything at a distance will be considered unlabelled*. There is also an option to take an image of the *top view* of the scene. Scenes have been created with the help of **Procedural Terrain Painter**, among a few other free packages. 1,000 Datapoints had been generated for model training and validation.

Training and testing model

The **AI4Mars** dataset has been downloaded and extracted to be used only when the model is finalised. After the Unity simulated data is ready, the jupyter notebook *training-and-testing.ipynb* is run. It was observed that the colour mapping in the labels was different, therefore mapping dictionaries and a function was created accordingly. Remaining steps taken, can be seen in the notebook; a Unet model is defined, trained on unity synthetic data, and tested on AI4Mars data.

Results and findings

Results are as below

```
Samples evaluated: 16386
Mean IoU: 0.1002
Pixel Accuracy: 0.2333
```

While the pipeline for simulating and for training and testing appear to have been set up properly, the results don't look like the current model can be of use. Improvement is

Future Work and Improvements

- Rover: One thing to do is to add a rover and include it in the *unlabelled* part. This is not there in the synthetic dataset, but is a significant part of the real one.
- Expand: One option would be to train on a larger dataset and with more epochs. However, sometimes a small model trained well can be just as good, and particularly with synthetic data, there might be some limits on this.
- Orbiter dataset: As mentioned above, already there is a provision to take a top view synthetic images as well. Without looking at the *lander/rover* dataset, it should be possible to compare these synthetic images with real images taken from *orbiter spacecrafts* and improve the simulation based on them. This may improve the performance significantly. I believe it is definitely worth experimenting with.

GitHub Code:

Link:

https://github.com/varun-analyst/CS6886-proj-Unity-Mars-Simulation

There is a *readme.md*, and also two more in each of the subfolders which can be read for reproducing the results.