Martian Terrain Height Estimation from Single Satellite Images with CNNs

Juliana Chew
MIT
77 Massachusetts Avenue
jlchew@mit.edu

Alexander Koenig
MIT
77 Massachusetts Avenue
koe@mit.edu

Abstract

We design a height detection deep learning algorithm for Mars satellite terrain imagery via singular-perspective satellite imagery. Existing methods currently in use in space use stereo imaging, synthetic aperture radar, or laser interferometry, so a method using only one camera can enable instantaneous and relatively low-SWaP terrain estimation. This algorithm may be useful where (a) taking stereo imagery is not feasible (for example, when the spacecraft location is static), and (b) where no existing height maps exist, as would be the case for exploration of unfamiliar planetary bodies.

We trained a UNet with the dataset generated by the HiRISE imaging camera on the Mars Reconaissance Orbiter, which has approximately 7000 stereo image pairs and corresponding terrain maps. We used a UNet because of its ability to retain both spatial context and features and its encoder-decoder structure, skipping the traditional SoftMax layer to perform pixel-wise regression. Our model has poor performance overall, generally producing highly flat yet noisy height predictions regardless of the actual terrain. However, its height predictions do somewhat correlate with actual image features such as valleys and craters, and the model sometimes fails in meaningful ways such as mistaking craters for mounds and vice-versa.

1. Introduction

The exploration of Mars and other planetary bodies has been a priority for the current frontier of space exploration. To date, unmanned missions have required known height maps to preselect landing areas — for example, with the Perseverance rover, the Mars Reconaissance Orbiter was used to determine Martian topology, which was then hand-selected for landing regions of interest. To autonomously choose landing areas for planetary bodies with unknown topology, however, a more automated process is required.

Stereo imagery, synthetic aperture radar (SAR), and LiDAR have been popular methods to analyze planetary surfaces from space. However, these methods often involve a higher SWaP (size, weight, and power) because stereo imaging involves multiple cameras (or longer timescales), and SAR and LiDAR involve more expensive instruments with higher power requirements. Therefore, a depth-estimation method using solely one camera would reduce not only SWaP but also cost, making this approach useful for real-time terrain estimation when other methods are infeasible.

2. Related Work

2.1. Classification vs Regression

As Mou and Zhu [4] note, terrain height estimation from monocular imagery is a relatively underexplored capability within the remote sensing community. Previous work on height estimation from satellite or aerial imagery has taken one of two broad approaches: height estimation is either treated as a continuous regression problem, as in Mou and Zhu [4], or height intervals are discretized into different classes and height estimation is treated as a classification problem, as in Li *et al*. [3]. Some approaches have used both methods simultaneously, as in Srivastava *et al*. [6].

In our model, we used regression to create a continuous rather than discretized output height map. This continuous output is useful for Mars remote sensing, particularly when finding potential landing sites, as the depth ambiguity in height discretization can be dangerous for a rover or other space instruments. Furthermore, classification is more applicable to (and, in literature, was used more frequently for) manmade features such as buildings and other structures, where discretized height makes more sense — e.g., one may be interested in determining whether a section of the image is a "tall building", "short building", or "no building". For non-manmade features, which are more continuous in nature, the classification approach is less suitable.

A downside of performing regression is that it poses a more difficult technical problem, particularly since 6.819 centered on classification networks. Taking inspiration from the successful implementation of a regression ap-

proach in Mou and Zhu [4] and Chen *et al*. [1], however, we decided to take the more ambitious path, especially as it is more suitable for the problem at hand.

A regression-based approach specifically for Martian terrain has been successfully implemented before by Chen *et al*. [1], who note the particularly strong challenge that Martian terrain estimation poses compared to other planetary bodies (such as the Earth or moon) due to instrumentation noise, difficult-to-characterize atmospheric affects, and Mars' surface albedo fluctuations (i.e., highly variational soil reflectivity). To counter these specific challenges, they split their approach into two subnetworks. The first network focused solely on calibrating the input images by performing denoising, correcting the surface albedo variations across the image, and re-illuminating the scene with a consistent illumination angle. The second network was a CNN which solely performed the depth estimation task.

For our method, we made an intentional decision to not take the two-network approach as in Chen *et al*. [1]. Their method relied on the Mars Express dataset rather than the Mars Reconaissance Orbiter (MRO) HiRISE dataset. Mars Express was launched several years prior to the MRO, and in comparison, the MRO uses highly exquisite and well-calibrated instrumentation (it is the largest and most high-resolution imaging system ever operated outside of Earth orbit). Images from HiRISE are also extensively precalibrated by the NASA JPL science team prior to public release — although they are not re-illuminated with a consistent illumination angle — whereas the public Mars Express dataset has relatively minimal calibration. As a result of this higher level of calibration and lower noise overall, we believed it possible to successfully execute a single-network approach. This choice also brought the problem within the scope of a course project, as either of the subnetworks would comprise an entire project on their own. The ramifications of this shortcut we took ended up being quite significant, and are discussed in detail in Section 3.5.1.

2.2. Loss Function

Eigen *et al*. [2] developed a new error metric for depth perception, citing that a scene's global scale was not only a fundamental ambiguity but also a large contributor to current depth perception errors. Compared to RMSE (as used in most other approaches), scale invariant error had a 20% better performance due to its ability to gauge the relative difference between corresponding points without regarding differences in global scale. As global scale was indeed variable in our dataset (the pictures were often taken at different heights above Mars' terrain), we opted to use scale-invariant error to improve performance. For the mathematical definitions of scale invariant error, refer to 3.4.1.

3. Method

3.1. Data Acquisition

We developed and utilized a web-scraper to extract relevant stereo pairs and terrain maps from the HiRISE website[1] and converted the various formats to tensors. As the dataset is ∼250 GB, we started an AWS instance to download and store the data in a S3 bucket for later use. That way, the data are accessible across multiple instances.

This dataset is not perfect in its height predictions, generally having root-mean-square errors on the order of a fraction of a meter to several meters depending on the exact terrain map. This places a negligible bound on expected model performance, but in general the dataset is highly accurate however and certainly sufficient for use in training such a model.

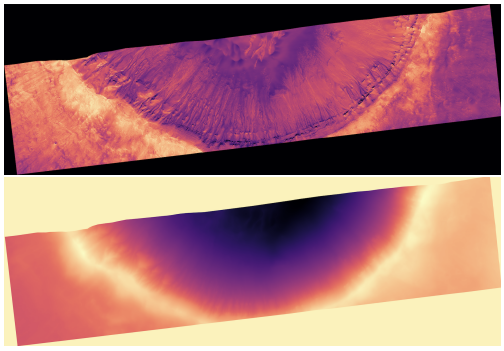

Figure 1. An example HiRISE image and its corresponding altimetry map. False color shows terrain height.

3.1.1 Downselecting Data Formats

We determined how we will select our training data from the available data formats, listed in Table 1. While diversity in the training data formats is beneficial for training, some formats must be consistent: to ensure a consistent elevation definition across the data set, we used only areoid rather than radial elevation. Similarly, we downselected from color images to monochromatic images because (a) a model cannot easily simultaneously handle several-channel and single-channel inputs, (b) 90% of the dataset was given in the red channel, and (c) soil albedo — which we desired to keep as consistent as possible — highly depends on image wavelength.

[1] HiRISE PDS, https://hirise-pds.lpl.arizona.edu/PDS/

Format Category	Available Types
Color	Color (IR, red, blue), Mono (red)
Pixel resolution	0.25m, 0.5m, 1.0m, 2.0m
Map projection	Equirectangular, Polar Stereographic
Elevation Profile	Areoid, Radial

Table 1. Available data formats within the HiRISE data set.

3.2. Data Augmentation

As HiRISE images are often thousands of pixels wide, we originally cropped the images and targets to square patches that were 256 pixels wide to reduce memory usage and computation time. However, we found that cropping small patches proved detrimental to the model's performance because the model was not provided enough overall spatial context to determine depths.

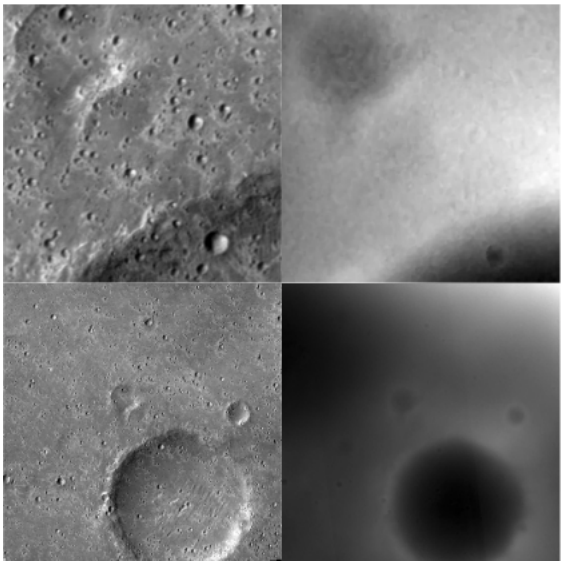

Figure 2. An example of the affects of randomly cropping an image. The camera image is in the first column, and the target altimetry the second. The first and second row were randomly cropped with patches 256 pixels and 1024 pixels wide, respectively. The larger cropping provides more terrain context (i.e. it is apparent there is a crater). In the smaller cropping, though, it is difficult to sense the large crater.

To provide more terrain information, we cropped the images and targets into random 1024-wide patches, then downsampling them by 4 times to produce an image-target pair that was 256 pixels wide. By doing so, we provide more terrain context while still maintaining a reduced computation time.

3.2.1 Normalizing the Height Maps

The HiRISE height maps are given in areoid elevation, and therefore specify not the height with respect to the mean of a particular image, but rather the absolute height relative to the Mars areoid (essentially, the equivalent of Earth's mean sea level). We did not expect our model to be able to learn the absolute height of the image, however, as that would require being able to identify the overall region of Mars which the individual image belongs to. Therefore, we normalized the height map of the cropped images to have a mean of 0 by subtracting the height mean from every pixel in the image. In order to enable the model to distinguish relatively flat terrain from hilly terrain, no other scaling normalization was performed, i.e., the difference between the minimum and maximum height in each image was not altered in any way.

3.2.2 Dealing with NaNs

The HiRISE dataset assigns the value $-3.38\text{E}38$ as a placeholder for invalid and nonexistent data. Because of this assignment, the losses and gradients quickly exploded during training. We initially set all occurrences of $-3.38\text{E}38$ to 0 to counter this phenomenon, following Chen *et al.*'s method [1]. However, this approach produced artifacts along the edges of the model's layers, as the model quickly learned that the value 0 most likely occurred along image boundaries.

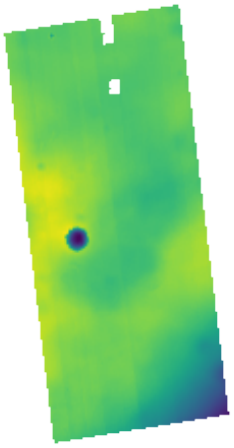

Figure 3. An example of an a HiRISE altimetry image with missing data, shown in white.

Therefore, we opted to randomly crop the images and targets, only continuing the augmentation process (downsampling, random horizontal and vertical flips) until no extracted pixel value was $-3.38\text{E}38$. The median number of random crops for each image to produce a usable sample was 1, and the mean was 1.3, suggesting that overall this

approach only excluded a small portion of the dataset for use in training.

3.3. Our Model

We started with an encoder-decoder structure, where the encoder's backbone was a Resnet50. As discussed in Office Hours, Resnet50 was likely not the best model to use for pixel-wise regression as it cannot retain high-resolution information for the image. Therefore, we changed our model to a Unet to perform pixel-wise regression.

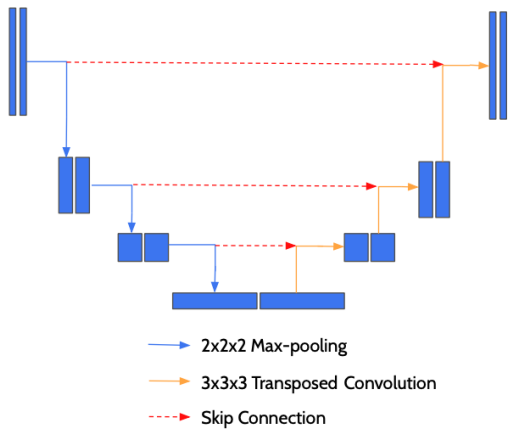

Figure 4. Our UNet Model. The input and output size was 8x1x256x256, where the training batch size was 8. Adapted from Schmidt [5].

We chose a UNet because of its use in remote sensing for semantic segmentation and its encoder-decoder structure. As it is able to extract features while retaining spatial information, UNet was an appropriate choice for pixel-wise regression. To convert the UNet from a classification to a regression name, we did not use a SoftMax layer traditionally used after the UNet.

3.4. Training

We randomly assigned image-target pairs into one of training, validation, and test datasets. The training set comprised 80% of the images, validation 10%, and test 10%.

3.4.1 Loss and Optimizer

For training, we used SGD (learning rate as 1E-7) with scale-invariant MSE as our loss function [2]:

$$D_{L2}(y, y^*) = \frac{1}{n}\sum_{i=1}^{n}(\log y_i - \log y_i^*)^2 \quad (1)$$

$$D_{SI}(y, y^*) = \frac{1}{n}\sum_{i=1}^{n}(\log y_i - \log y_i^* + \alpha(y, y^*))^2 \quad (2)$$

$$\text{with } \alpha(y, y^*) = \frac{1}{n}\sum_{i=1}^{n}(\log y_i - \log y_i^*)$$

Thus, our combined loss function J is defined as follows:

$$J = \lambda D_{L2}(y, y^*) + (1-\lambda)D_{SI}(y, y^*) \quad (3)$$

In this function, $\lambda = 0.5$, as it was found by Eigen *et al.* that this value produced accurate prediction and improved the qualitative result [2]. By using this loss function, we account for global scale ambiguities between the model output and the target. Not accounting for scale ambiguities can be detrimental to performance because small objects close to the camera can appear identical to larger objects that are further away. In this situation, the model will have difficulty determining if a given feature is shallow or deep. In our loss function, $\alpha(y, y^*)$ calculates the average scale difference between the two images, in effect "converting" the images to a common scale and comparing corresponding pixels' depths relative to their respective images.

3.5. Evaluation

Overall Performance

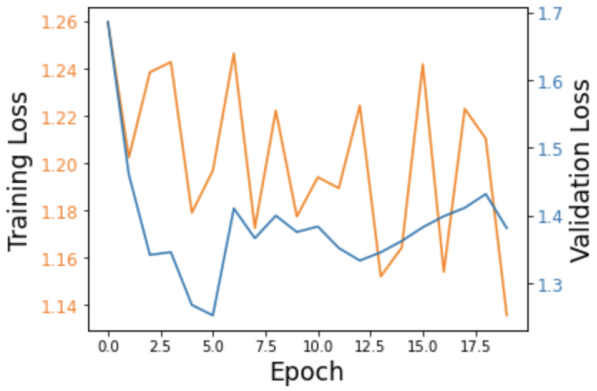

Figure 5. The training (orange) and validation (blue) loss curves of our model using standard invariant error. The training error fluctuated a lot due to the smaller batch size. Having a larger batch size would mitigate this problem, but we encountered memory issues with larger batch sizes. The model appears to perform best at epoch 5 and overfit afterwards.

As shown in Figure 5, the performance of our model left much to be desired. Unfortunately, the log validation loss was approximately 1.25 at best.

The model was unable to discern large, overall features, mainly only able to detect depth changes when there was a high intensity gradient.

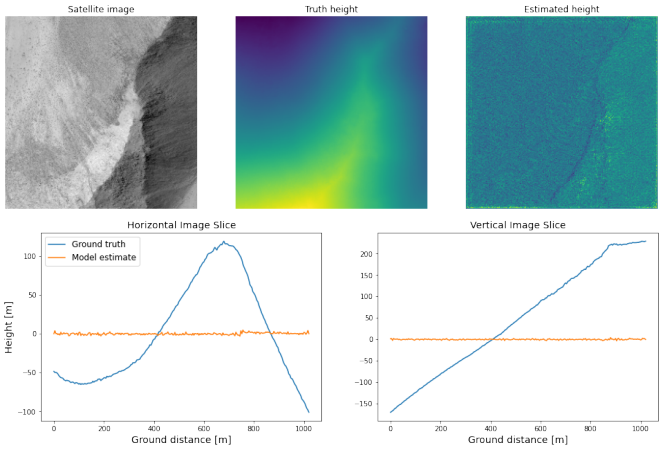

Figure 6. An example model output and target for a ridge on Mars, shown on the first row. The horizontal and vertical slices are along the middle of their respective axes.

As shown in the figure above, the model output (in orange) appears to poorly match the ground truth, instead predicting a relatively flat terrain regardless of input.

However, we trained this model with scale invariant error, as mentioned in Section 3.4.1. Therefore, a more informative comparison between the model output and ground truth should also be blind to global scale. In the figure below, the predicted height map is rescaled to better compare it to the truth height map. Since the model output was rather noisy, especially after being rescaled, a moving average was also taken across 20 pixels.

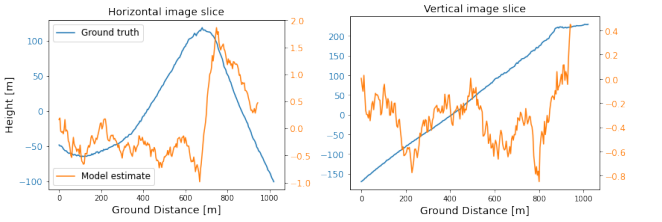

Figure 7. The output and target for the same ridge on Mars as Figure 6, only with different scales.

As shown in Figure 7, the model was able to detect a sharp change in elevation around the ground truth peak in the horizontal slice, albeit detecting the peak later than it actually occurs. This behavior is likely due to the sharp gradient in the input image in 6, where a line in the middle of the peak (likely due to different sediment compositions) occurs.

In the vertical slice, the model performed less well. Although the prediction did show an increase in elevation, the high fluctuation, "mini" peaks, and abrupt changes do not reflect the ground truth trend well. Like in the horizontal case, this behavior is likely due to ambiguous lighting and varying sediment colors, which are discussed in section 3.5.1.

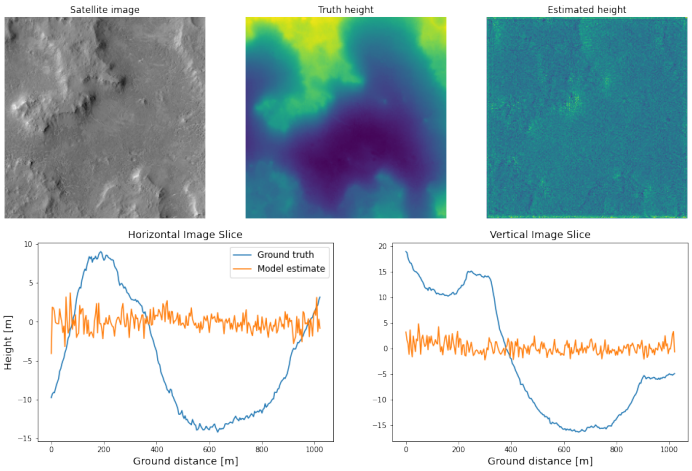

Figure 8. An example model output for a valley in Arabia Terra. The model output is rather noisy, and is essentially flat compared to the actual terrain.

As before, the height map is rescaled and smoothened to produce the plot below, which shows a different comparison between the model's output and ground truth for this particular image.

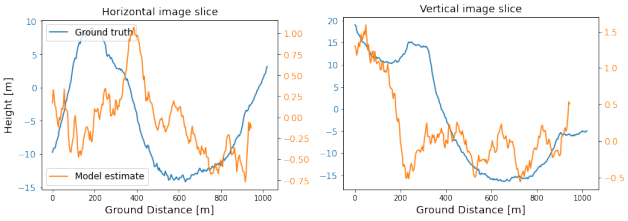

Figure 9. The rescaled output compared to ground truth height. The height maps begin to *vaguely* match; the predicted height drops at/near the valley floor, and rises at the edges of the valley. Although we did not hunt for better-than-average examples of model performance to present here, it is worthy to note this is perhaps the best performance we saw across all viewed examples.

This particular example shows in 9 displays well that the model is on the right track to performing reasonable terrain prediction, despite that it has a long ways to go. The model is able to recognize the presence of a valley within the image, and also displays correctly a rise in height at the plateaus surrounding the valley. It is important to emphasize that this success should be taken with a healthy amount of skepticism, since many other model outputs did not achieve even this low level of performance.

Within the next few image sets, note that for brevity, only the rescaled height maps are shown along with the image sets.

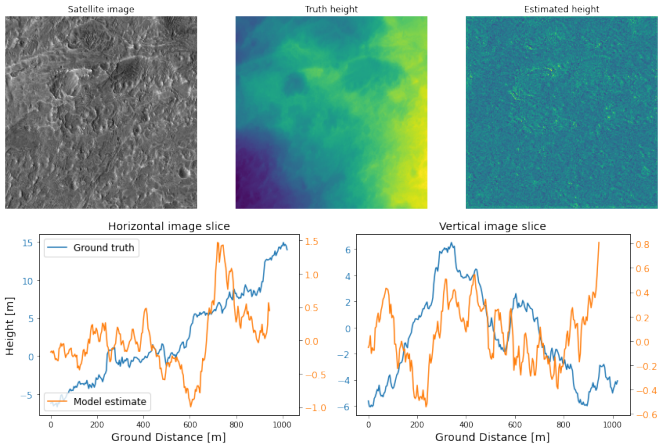

Figure 10. An example model output for a region with gently sloping terrain (a subsection of the Roddy Crater in Arabia Terra). Since the model already outputs mostly-flat predictions, the prediction matches ground truth somewhat well. This example displays one of the model's better prediction successes.

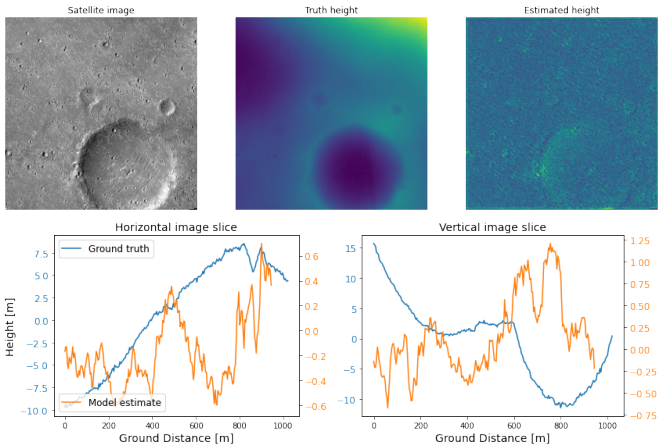

Figure 11. Model performance on a large crater just south of Elysium Planitia. The model makes a prediction about the crater height, but the wrong one: it believes it to be a mound rather than a depression. This behavior is expected if the model thought the scene was lit from the bottom right of the image rather than the top left, as it actually is. Similarly, for the horizontal image slice, it predicts a crater rather than a mound, which further evidences the hypothesis that it made a faulty prediction about illumination angle.

Since this model was built with the intent of being usable for exploration of unknown bodies, Figure 12 shows model performance on a sample moon image, since the model has never previously been shown lunar terrain during training. The truth height for this image is not known (this image was taken by an author of this paper years ago), but nevertheless, model performance can be evaluated qualitatively. The model performs about as poorly as it did on Martian images, but perhaps no worse, either. It successfully picks up on small hills within the center of craters. It mistakes two small craters for a hills on the left and bottom of the image, and generally does not pick up well on large-scale image features such as the two large craters.

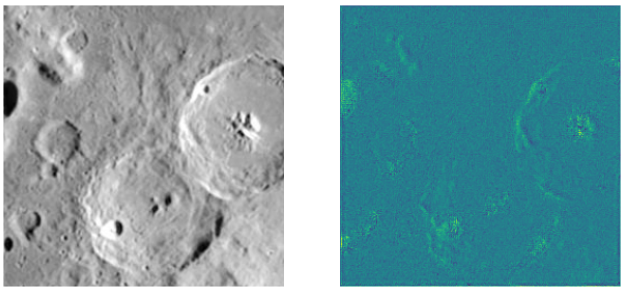

Figure 12. The model's output on a sample lunar image with an undetermined height map.

3.5.1 Lighting and Sediment Composition

As light is instrumental for depth perception, a confusing or ambiguous lighting can prove detrimental for prediction performance. This performance effect appears to be applicable to our model in our investigation; the model frequently predicted actual height drops as height increases and vice-versa. We expected to be able to rely on the low noise and high level of calibration of the HiRISE dataset to alleviate lighting-related issues, but this expectation proved ill-judged: although the dataset is highly calibrated with regards to instrumentation noise and Mars atmospheric affects, affects from surface composition and different illumination angles are still strongly present within the data.

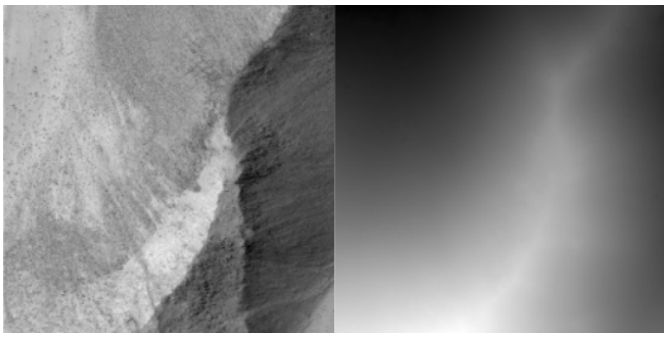

Figure 13. A cliff on Mars, where the HiRISE image (for red color) is on the left, and the altimetry on the right. The intensities in the input image makes depth perception difficult–the image may appear to be a ridge, but is actually a cliff where the darker left side has less intensely red sediment.

As shown in Figure 13, the lighting in these aerial images make perceived depth ambiguous. Much of related depth

estimation work used day-to-day images on Earth, where shadows were highly indicative of both shape and depth. However, as the HiRISE images were taken in varying times of Mars day, shadows (or the lack thereof) can make shapes ambiguous. For instance, if a picture were taken when the Sun were angled away from the imaged site, then shadows would be useful in determining depth. However, if the Sun were directly overhead, most (if not all) shadows would disappear, leaving the terrain highly illuminated but with little context for depth perception. Even more critical is that if the illumination angle is in the opposite direction, the meaning of shadows and bright regions are flipped; i.e., a shadow in one illumination angle may correspond a drop in terrain height in a certain direction, whereas a shadow in the opposite angle would instead correspond to a rise in terrain height.

Another ambiguity comes from sediment composition. Our input images measured only the red intensity, which, for the red planet Mars, seemed to be an appropriate choice. (We also chose to only use the red data because other wavelengths' data were not as prevalent.) However, due to variations in composition, the intensity of red in the sediment can vary. For instance, less red areas may appear darker in the image, and more red ones appear lighter. In the first case, the model may predict the terrain to be deeper than anticipated, while in the second case, it may predict it to be shallower.

4. Conclusion

Overall, our method did not perform as well as we hoped. Instead of predicting terrain heights, it generally predicted relatively flat yet noisy terrain, only detecting smaller features (such as small craters or hills) if they had a sharp luminosity gradient in the input image. When looking at rescaled and smoothed height predictions, it is clear that the model is taking correct initial steps towards accurate terrain prediction, but still has a long way to go. In some select cases, it was able to detect valleys and ridges and, in a low-accurate and noisy manner, associate those features with a meaningful height prediction. Where the model fails, it does sometimes fail in meaningful ways, such as mistaking craters for mounds (likely because it believes the scene illumination angle is opposite from the actual angle).

Unlike Chen *et al*. [1]'s approach, we did not use a dual-network approach to separately re-illuminate / calibrate images and then perform height regression, instead aiming to have the UNet itself implicitly determine (and account for) illumination angle and surface albedo variations. In the end, this approach did not succeed: it was apparent that our model frequently mispredicted height drops as height increases and vice-versa, which would have been resolved via image re-illumination. In hindsight, we believe this was likely a primary cause behind the failure of our model.

Beyond moving to a dual-network approach to unify illumination angles and albedo variations across the dataset, there are further ideas on how to make the model functional. One way the model could be improved is to use an ensemble approach, where one model detects large, overall features (such as valleys, cliffs, or large craters), and another detects smaller ones (such as small ridges, holes, and bumps). This can be achieved by using identical models (such as a UNet, CNN, etc.), where one takes random small patches of the images as input and the other uses a downsampled version of the entire image. The results from both models can be combined, either by pure addition or by a weighted average, to produce a terrain depth map.

5. Individual Contribution

Within the technical side of this project I primarily managed the data-wrangling component (reading HiRISE documentation, determining how to load NASA-specific image formats, reading about the dataset limitations and formats, figuring out what selection of the dataset to download, handling the AWS S3 bucket) and data analysis (looking at model outputs, analyzing individual layer activation, coming up with new ways to visualize model performance such as taking height "slices" of the image). Both of us trained various (and many!) models. Just like the project itself, the report was also highly collaborative — we each essentially worked on all the sections. If any, the ones I would take 'primary' credit for are the related work section and generating images/plots for evaluation, although I also wrote content for the abstract, intro, dataset usage, evaluation, and conclusion.

References

[1] Z. Chen, B. Wu, and W. Chung Liu. Deep learning for 3d reconstruction of the martian surface using monocular images: a first glance. *ISPRS*, 2020. 2, 3, 7

[2] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. *Adv. Neural Inform. Process. Syst.* 2, 4

[3] Xiang Li, Mingyang Wang, and Yi Fang. Height estimation from single aerial images using a deep ordinal regression network. *IEEE*, 13(9):1–5, 2006. 1

[4] Lichao Mou and Xiao Xiang Zhu. Im2height: Height estimation from single monocular imagery via fully residual convolutional-deconvolutional network. *IEEE*, 2017. 1, 2

[5] Johannes Schmidt. Creating and trainng a u-neet model with pytorch for 2d 3d semantic segmentation: Model building [2/4], 2020. https://towardsdatascience.com/creating-and-training-a-u-net-model-with-pytorch-for-2d-3d-semantic-segmentation-model-building-6ab09d6a0862. 4

[6] S. Srivastava, M. Volpi, and D. Tuia. Joint height estimation and semantic labeling of monocular aerial images with cnns. *IEEE*, 2017. 1