EFFICACY OF GENERALIZING NASA'S HLS GEOSPATIAL FOUNDATION MODEL FOR DOWNSTREAM APPLICATIONS ON MARS

Safi Patel, Emos Ker, Eugene Chang & Sanho Lee
New York University
CSCI-GA 2565 Machine Learning
{sp6559, ck3189, ec4338, shl8607}@nyu.edu

ABSTRACT

The Prithvi-100M model is a foundation model created for geospatial tasks after being trained on Earth satellite data. It performs remarkably well on finetuned tasks on Earth, with the original researchers showing impressive results with minmal data. Our study aims to push this generalizability to see if it could work on Mars. We attempted a very similar segmentation task as the original researchers by attempting to train a model that could segment craters in Martian images. After thorough experiments we concluded that the foundation model was not generalizable enough as we first thought to train a finetuned model with a comparable segmentation task on a comparable amount of data. However, we were able to successfully finetune for a different and noncomparable classification task with an ending 90.01% validation accuracy. This success, along with the our insights regarding a better and impressive imputation method led us to conclude that while this foundation model may be limited in its generalizability than first hypothesized, there is still great potential to use this foundation model for downstream tasks in a way that could still benefit planetary science.

1 INTRODUCTION

Foundation models and for transfer learning have shown in recent years impressive benefits in terms of robustness and generalizability compared to creating specialized datasets and training from scratch. The Prithvi-100M model, developed collaboratively by NASA and IBM Research is one such foundation mode. Pre-trained on petabytes of Earth satellite imagery sourced from NASA's Harmonized Landsat Sentinel-2 (HLS) dataset, the researchers aim to provide a general model for geospatial tasks. By adopting a masked autoencoder (MAE) approach built upon the vision transformer (ViT) architecture, Prithvi-100M aims to deliver state-of-the-art results across numerous geospatial downstream tasks with minimal fine-tuning. Examples of downstream tasks include flood detection using the Sentinel-2 data from the Sen1Floods11 dataset, burn scars detection using the NASA HLS fire scars dataset, and multi-temporal crop classification using the NASA HLS multi-temporal crop classification dataset. They showed that with a single foundation model, they could use fine-tuning for a wide array of seemingly disparate tasks that traditionally require purpose-building and curating data in a tedious and sometimes ineffective fashion. However, could we utilize that same efficacy and generalizability to apply to the geospatial features of another planet? In this paper, we would like to see how far we can push this idea of generalizability by evaluating the effectiveness of a fine-tuned HLS model on downstream applications on satellite imagery of Mars, such as a comparable segmentation task of Mars craters.

If we show the fine-tuned HLS model as effective and reliable even on Mars imagery, it will have implications for computer vision foundation models in general, by giving further evidence and credibility to the idea that they can act as excellent generalists. More importantly, we will have shown that future planetary science utilizing geospatial data doesn't have to rely on purpose-built models or solely on single-planet data. Instead, those researchers can utilize, rely, and benefit from the sheer scale and robustness that the foundation model approach like HLS can provide.

Central Research Question: Can the NASA HLS geospatial foundation model that has only been trained over Earth data and has been shown to be finetuned on downstream segmentation applications, be generalizable and robust enough to effectively perform a comparable segmentation task on Mars with a comparable amount of training data?

2 Related Works

2.1 ViTMAE

The Prithvi-100M model incorporates the Masked Autoencoder (MAE) approach with a Visual Transformer (ViT) architecture, a concept introduced by He et al.(1) that focuses on pre-training transformers to reconstruct images from masked input patches. The HLS Foundation model uses the MAE training strategy but augments it to make a ViT architecture that replaces the 2D patch and positional embeddings with 3D embeddings to handle the temporal dimension, an attribute that is first of its kind. However, we don't utilize this uniqueness for this task and instead keep our experiments with $T = 1$ time frame to match the researchers' fine-tuning examples that also made the same decision.

2.2 Applications Beyond Earth

In planetary science, the use of AI models has traditionally been constrained to specific datasets. Approaches like automatic crater detection (2) and terrain classification (3)(6) have demonstrated the value of machine learning for planetary exploration. However, these are bespoke models with their own datasets and own training disconnected from one another, which limits its applicability to a specific celestial body. We want to show that foundation models for geospatial tasks can also be used for other solar system objects, which would then spread the benefit from scale, robustness, and performance of foundation models throughout planetary science.

2.3 Prithvi-100M Previous Usage

Little research has been published utilizing the Prithvi-100M model because of the relative recency of its release. Blumenstiel et al. (4) discusses the use of deep learning models for efficient image retrieval from large satellite image datasets without requiring annotations. Their work proposes using Geospatial Foundation Models, such as Prithvi, for remote sensing image retrieval, highlighting two main advantages: encoding multi-spectral satellite data and generalizing without additional fine-tuning. Relevant to this research, they showed an example of utilizing the foundation model with only 3 RGB channels even though the model was trained on and expects 6 channels. We compare their method of imputation with other, more effective methods in later sections as an intermediate result.

3 Approach

3.1 Image collection and Segmentation

The publicly available HiRISE images as taken by NASA's Mars Reconnaissance Orbiter provided our collection of high resolution Martian imagery for training data. To conform to the size required by our segmentation pipeline, images were cropped into sections of size 512×512 containing identifiable craters. Extracted images were then manually segmented for craters using Roboflow, a conversion tool for computer vision datasets. To compensate for the lack of experience in crater segmentation, we defined craters to be a clear indentation on the Martian floor identifiable through an outline. Craters that were visibly identifiable but too small to be captured through the segmentation tool were ignored to reduce noise. By the end of the process, there were a total of 335 segmented images.

The model normalization parameters that are provided along with the pre-trained model for use are given in terms of **reflectance units**. For example, the means of the six channels are: 775.2, 1081.0, 1228.6, 2497.2, 2204.2, and 1610.8. This is because the original training data are from satellites that

use reflectance units and provide images in the GeoTIFF format, rather than standard image formats using a data range of 0-255 (uint8) or 0-1 (float). Therefore, when we load in our custom Martian dataset (a standard .jpg RGB image), we have to convert each channel to be on the same scale as the reflectance units. To that end, we multiplied each ith channel by a factor of

$$\frac{[i\text{th channel mean}] + [i\text{th channel std}] \times 2}{255}$$

Rather than scaling based on the complete possible range that reflectance units can have (1-10,000), we instead scale based on the range that most imagery actually falls under.

3.2 Image Reconstruction

A key issue with this approach is that unlike the images used for the original Prithvi-100M model which contained 6 channels, the images from the HiRISE dataset only contained the 3 RGB channels. The architecture of the pre-trained model cannot accommodate these images because of the mismatch in shape. To address this, we impute the remaining 3 out of 6 channels. At the same time, we want to maximize the amount of information we have from our images. Therefore, we don't want to prioritize simplicity and just duplicate the 1st channel for all 6 channels. Instead, we seek to utilize all the original channels of our new imagery. We acknowledge a few different reasonable approaches to this problem.

1. Blumenstiel et al. (4) used the "means" method, where they just fill the three channels with their respective normalization means, resulting in the 3 imputed channels showing all zeros post normalization.
2. Another method is pre-normalization duplication, where we simply duplicate the 3 RGB channels as the last 3 channels after scaling into the reflectance units. The Pre-normalization duplicated method involves duplicating the RGB channels into the last 3 bands before performing normalization on all channels. Post-normalization duplication copies the normalized RGB channels to the last 3 bands. The means method fills the last 3 bands with the means of the RGB channels. After normalization, the duplicated bands with the means becomes zeros. After imputing the bands, the images were passed into the the model with the original pipeline provided by the model.
3. The third method is the post-normalization duplication, where we also duplicate our 3 RGB channels, but this time after the normalization is already done.

To evaluate these methods we put a sampling of our Martian imagery (n = 259 **through the unfine-tuned Prithvi-100M model** with 0.5 masking ratio and **evaluated its reconstruction both qualitatively and quantitatively** (MSE loss over RGB channels).

3.3 Segmentation Task

With the preprocessed Martian crater dataset, we used the researchers' example of segmentation finetuning tasks as a template for our segmentation task. We used a very similar setup for our tooling, data pipeline, and finetuning model architecture. All experiments were run on the 4 v100 GPUs partition on NYU's HPC Burst servers.

For segmentation, we utilized the MMSegmentation tool created by OpenMMLab. This is the same tool utilized by the original researchers in the example finetuning tasks like flooding/burn scar area segmentation. The general setup can be found as configurations in the code files. We detail the parts that remained constant throughout all experimentation.

Pipeline. We first **load the image and annotation** using the imputation method using the findings and descriptions above (scaling to account for reflectance units and then duplicating the channels post-normalization to get 6 bands). Next, we use a random flip with probability of 0.5 in order to add augmentation. Finally, we use a random crop in order to take a 224x224 image and annotation title from the larger 512x512 image.

Finetuning architecture. We first apply the augmented ViT as an encoder. We use patch sizes of 16 to get a total of 14 patches and embeddings sized 768. Next, we have a "neck" of 4 of the PyTorch ConvTranspose2D layers, which act like "deconvolutions." We finally have a fully-connected

Conv2D layer to act as a per-pixel classifier.

Many MMSegmentation Experiments. Because of the many poor results for training, we had to run many different experiments. We started with a learning rate of 1.5e-5 and similar hyperparameters to the original researchers' burn scar detection application. We also started with Dice loss, which is used often for highly imbalanced segmentation tasks. In all, we evaluated more than 30 experiments. We tried changing and adjusting many combinations and permutations of the loss function (dice vs. weighted cross entropy), learning rate, warmup rate, class weights for cross entropy, the dropout rate, and batch size. We aimed to optimize for the validation set's mIoU metric (mean Intersection over Union), the average of the measure of how well the model segments a certain class.

Crater outlining. While we adjusted hyperparameters and loss functions, one of the major factors we changed is the way that we created our annotations in Roboflow. We originally started with the polygons from our annotations fully filled in with a crater classification. After this, we tried outlining them instead (as seen in Figure 1) to test if the outlines of the craters were the learnable and important features.

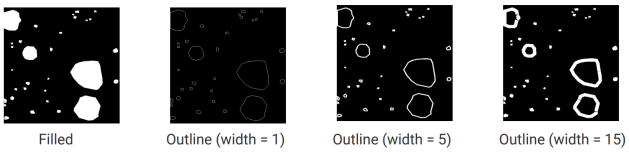

Figure 1: Examples of the same image with different ground truth methods

3.4 Classification Task

After it became clear that the segmentation task would not work on comparable data, we moved to a simpler classification task: **determine if a tile contains a crater or not**. For the classification task, we were no longer able to utilize MMSegmentation and had to completely overhaul the way the training was conducted. We utilized the Pytorch Lightning framework with a relatively similar setup.

New dataset. While we had decided to use a comparable amount of data ($n = 335$) images for the segmentation task, classification is a far simpler task with a higher risk of overfitting on the smaller dataset. Therefore, we more than doubled the dataset size (n=755) with an emphasis on reducing noise and outliers by choosing the craters that had no discrepancy between our human classifiers. At the same time, we made sure to keep the dataset diverse with a mix of colors, terrains, etc. to make sure our model was general enough for many kinds of Mars imagery. To accommodate all of this, more than 800+ HiRISE extended images were manually selected carefully. We hope that with this effort, this hand-selected dataset can be utilized for any further research regardless of whether they also focus on our specific area of foundation models, etc.

Fine-tuning architecture. The fine-tuning setup is considerably simpler than the segmentation task. First, we applied the augmented ViT as an encoder. Then, we extracted just the classification token, applied a Dropout layer, and applied a single Linear layer with an output to a single neuron. We applied binary cross entropy loss.

Masking as regularization. Due to the sheer number of model parameters involved, having twice the dataset size is not enough to prevent quick overfitting and significant degradation of the final validation performance. Note that we had existing "infrastructure" to easily apply masking noise to the image since the encoder was part of the larger class that was used during MAE training. Thus, we could simply **add a small probability of masking during training** to create an additional regularization effect besides the dropout layer.

Classification experiments. Similar to the segmentation task, we started with similar hyperparameters as the original researchers. However, we conducted more than 20 experiments where, again, we adjusted the learning rate, learning rate scheduling, warmup length, max training time, dropout probabilities, masking ratio for regularization, and batch sizes. We aimed to optimize the validation set's accuracy. We decided on this choice since our crater/non-crater balance was near perfect within the classification dataset, meaning accuracy was a good measure of our model's performance.

4 RESULTS

4.1 IMAGE RECONSTRUCTION

The image reconstruction using the pre-trained model was applied on a dataset of 259 images with a 0.5 mask ratio. We assessed the performance of the model using the model given loss and RGB channel loss, where the "model-given loss" is the mean squared error (MSE) across all channels whereas the RGB channel loss only considers the RGB channel MSE. Because our original Martian imagery is just in the RGB channels, we only care about reconstruction in these channels since the point of this is to see how well the model can grasp the latent information within the images.

We observed that for the "Means" method, the model-given loss was 0.0174 while for the RGB loss it was 0.0610. For the pre-norm duplication method, the model-given loss was 0.0298 and the RGB loss was 0.1040. Finally, the post-norm method showed a model-given loss of 0.0190 and an RGB loss of 0.0493. See Figure 2 below:

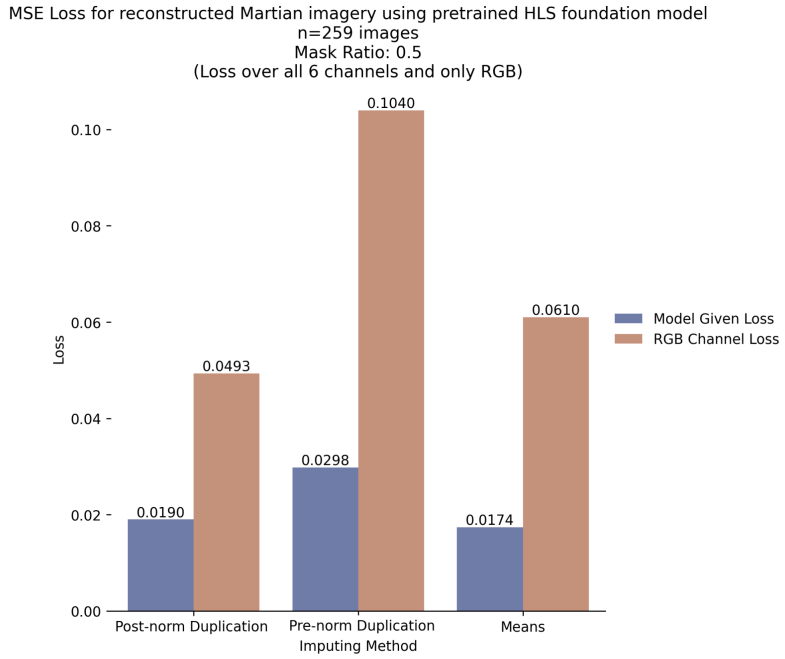

Figure 2: Model Given Loss and RGB Channel Loss for Post-Norm, Pre-Norm and Means method.

These quantitative results aligned with what we observed qualitatively. We can see an example in Figure 3. If we look at the pre-norm duplication method, the masked patches are readily apparent and they stick out as squares. The "means" method also has this characteristic and the discoloration is apparent. Only the post-norm method seems to blend in the texture and color of the rest of the image in a much more seamless way. Furthermore, in other examples, some of the hidden structures (eg. rocks, cliffs, craters) can be reconstructed to some degree. At worst, the model showed a blurry estimation of where there was previously some kind of structure. Throughout the testing, the model under the post-norm method did not demonstrate any hallucinations, false coloring, weird structures, etc. despite never seeing Mars imagery before during training.

4.2 SEGMENTATION TASK RESULTS

The results from our segmentation task were the main focus when investigating this research question and we found throughout all of our thorough experimentation that we were not able to train any effective models. Here are the highlighted results from a sampling of our many experiments:

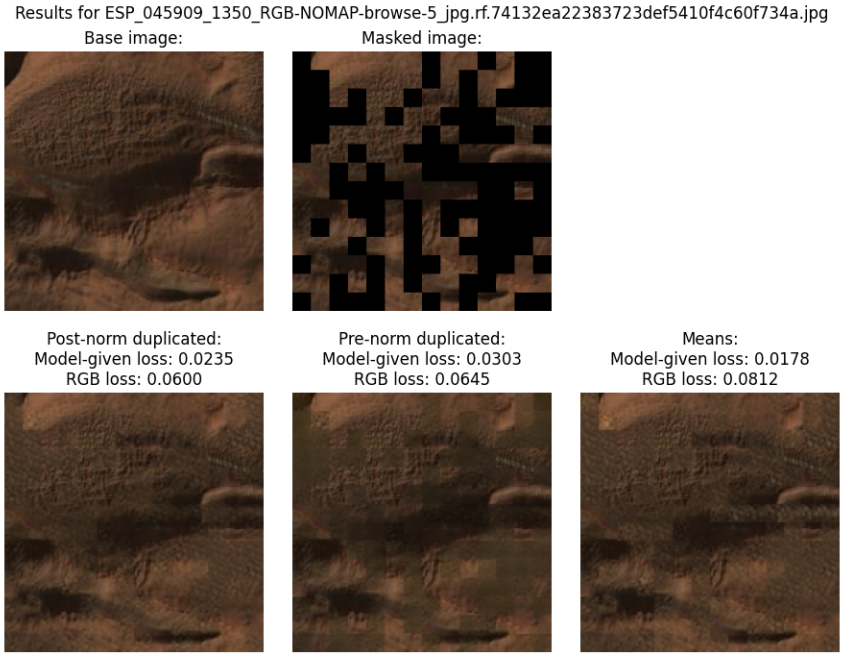

Figure 3: Example of pre-trained Prithvi-100M reconstruction of masked Mars imagery

Using dice loss and a filled crater annotation, we ended up with a training loss of 0.2079 but an mIoU of 0.4952. Note that because mIoU is the average of the two classes, seeing an mIoU close to 0.5 is most likely because the IoU of one of the classes is close to zero and one of them is 1. In our case, this was true for "non-crater", while the "crater" class was very close to 0.
When using weighted cross entropy (0.2 weight on "not within crater"/0.8 weight on "crater") resulted in a training loss of 0.1353 and an mIoU of 0.4629. Using weights 0.05/0.95 resulted in a training loss of 0.05687 and an mIoU of 0.2933, because in this case we saw that the IoU for even the non-crater class was thrown off, most likely because the vast differences in the weights caused unstable training. Using dice loss and the 5 width outlined annotations resulted in a training loss of 0.4859 and an mIoU of 0.493. Using dice loss and the 15 width outlined annotations resulted in a training loss of 0.4439 and an mIoU of 0.483. Using WCE (0.05/0.95) and the 15 width outlined annotations resulted in a training loss of 0.03793 and an mIoU of 0.3889.

And as far as qualitative views of any of these results, the finetuned model produced seemingly random spikes and stripes across the image that were essentially just noise and were not useful.

Figure 4: Example of an experiments model training results: WCE(0.2/0.8) - Filled annotations

4.3 Classification results

Compared to the previous segmentation task, when it came to the **classification task** we were able to train an effective model, **enough to declare the foundation model generalizable and successful on this classification task**.

Initial overfitting. At first, before we utilized masking as a regularization technique, we saw quick and significant overfitting no matter how we set up our experiment. In one of our experiments, we saw a textbook example with our training loss going to 0.0007724 but our validation loss rocketing up to 1.164.

Successful model. Our final winning setup with a learning rate of 8.2e6, dropout probability of 0.5, batch size of 4, and mask regularization ratio of 0.375 had a training loss of 0.0654, a validation loss of 0.3758, **and a validation accuracy of 90.01%.**We would like to note that higher masking ratios led to models that did not train well.

Figure 5: Training results for our final finetuned model for classification

5 Discussion

Imputation. Firstly, we believe that our insight surrounding the imputation is relevant to any researcher that ends up using the Prithvi-100M model. One thing to point out is the difference between the RGB loss and the model-given loss. We believe the reason behind this is because for some of the methods (eg. mean method), after normalization the last 3 channels are just zeros. Therefore, the model can have a very easy time predicting the masked areas within those channels as just zeros. In this way, the model-given loss is skewed and should be ignored as an effective measure compared to the RGB loss which is compared to the actual original image. Also, it makes sense that the post-normalization duplication method would work the best since that copies similar image data and maintains the same range since its after normalization.

We believe that this discovered method to better impute from standard RGB channels compared to the other researchers is backed up by our qualitative and quantitative results and should be used by any future research that handles RGB data and the Pritvi model.

Masking as regularization. We also believe that our use of masking as regularization is good support for its usage in certain contexts.

Previous research has shown that high masking rates (eg. ≥ 0.5) might lead to unstable training. Our success with a lower masking ratio (around 0.375) gives evidence to the idea that lower rates can be used effectively for training while significantly combatting overfitting (5).

However, we were not able to easily find using image masking as regularization for supervised learning of classification tasks specifically with a ViT architecture, and believe this to be yet another interesting and insightful finding.

Segmentation task. The negative results in regards to our segmentation task were surprising. We had initially seen the great, very successful reconstruction performance from the Prithvi-100M model and were impressed by its generalizability. This led us to assume that we would see comparable results for a comparable segmentation task if we used a similar amount of data as the researchers. However, this was not true and we failed to train any effective model even after thorough experimen-

tation. While it is hard to determine the exact cause without further research, we note a few likely possibilities and contributing factors.

5.1 Potential Limiting Factors for Segmentation Task

Lack of Data/Mars is inherently difference. Mars may just be too different in its structures, colors, terrain, texture, etc. in order for our model to learn the segmentation ideas with a comparable amount of data to an Earth task.
Crater Features. Unlike other geospatial features such as burn scars or floods, Martian craters might not have distinct, contrasting features that are easily distinguishable by a model. The absence of distinct color contrasts and reliance on subtle cues like shadows and shape make it challenging for the model to differentiate craters from other indents or geological formations.

The identification of craters involves interpreting circular outlines that could be ambiguous. These circular shapes might not always denote craters; they could represent other geological formations. This ambiguity can confuse the model, especially if it has not been trained with a sufficiently annotated and diverse dataset where such nuances are clearly labeled. Therefore it is possible that this task is simply not suited for this model architecture or would require a vast amount of data unlike the original researchers examples.

5.2 Future Research

However, our **success with the classification task** still shows promise for the HLS foundation model. It proves that while some tasks and amounts of data might not lead to success, there are routes that lead to generalizable models. These tasks should be actively sought out in order to bring the full benefit of the HLS foundation model to planetary science.
Given these challenges and limitations, further research would involve figuring out the reason for this lack of effectiveness even though we had a comparable amount of data and a seemingly comparable task. Was it because the task was ill-formed for this particular architecture or was it because there was not enough data? Answering these questions can unlock the limitations and expose the niche that this foundation model can help with in terms of aiding planetary science.

6 Conclusion

In this paper, we sought to explore how a state-of-the-art image machine learning model designed for and trained on Earth datasets, such as the Prithvi-100M model, may be generalized to images of Mars. We determined that a machine learning image model is capable of performing image reconstruction and classification tasks, but may fall short on segmentation tasks. We suggest that we observe the given results due to image reconstruction and classification being less prone to image details presented in craters of the Mars surface. The model's incapability to learn about segmentation of craters likely stems from the differing and detailed observations in other image features like shadows compared to the tasks of flood and burn scars which have clear color mismatches between regions. Either way, when it came to our original research question, we conclude that this foundation model was not as generalizable as we first hypothesized. It was unable to perform a comparable segmentation task on a comparable amount of data in the way that the original researchers did, either because of the nature of the specific task or because it would require more data.

In all, we hope that despite the negative results we observed for our main research question, our work along the way and the additional experiments conducted uncovered insights, methods, and concrete items for research in the future. The finding of the better imputation method compared to previous research can benefit future users of this foundation model regardless of their specific focus.

Our findings from our different experiments also act as additional support to using image masking as a regularization method, especially since we uniquely showed it in the context of a ViT architecture compared to previous research.

Finally, we had a great success with the classification finetuned model. This shows that there is still much promise in using this foundation model and generalizing it for the benefit of aiding planetary science tasks.

CODE REPOSITORY

All training code, utilities, notebooks, datasets can be found at our github repository:
`https://github.com/safipatel/martian-hls-foundation-model`

AUTHOR CONTRIBUTIONS

Safi was the main coder the project as he had the greatest knowledge of the Prithvi model and was well-versed in ML programs with Pytorch. For the same reason, he was also the reviewer of the paper contents as he had the strongest understanding of the results.

Emos, Eugene and Sanho focused on gathering the images from the source (HiRISE) by selecting suitable photos (e.g. non-grey scaled, non-false color), cropping and then segmenting them into crater sections. Then, the results from the model and training were left to this team to interpret and write the paper, providing suggestions on how the model can be improved.

Although the team of Emos, Eugene and Sanho wanted to contribute to the code, it was decided with deliberation that it was best handled by Safi (or generally 1 person) for a few reasons:

1. The image reconstruction and segmentation process was streamlined from the example code provided the model, which means only minor modifications were needed.
2. Safi was by far the most knowledgeable on the subject. Since the classification portion is streamlined from the previous part, he would have the best results.
3. Due to the linked nature of the tasks, assignment to additional hands would likely cause more disorganization and compatibility issues between code.

REFERENCES

[1] He et al. Masked Autoencoders Are Scalable Vision Learners. arXiv 2021.

[2] Palafox et al. Automated detection of geological landforms on Mars using Convolutional Neural Networks. Computers &Geosciences 101 2016.

[3] Stepinski et al. Automatic Recognition of Landforms on Mars Using Terrain Segmentation and Classification. Discovery Science, 9th International Conference, DS 2006

[4] Blumenstiel et al. Multi-Spectral Remote Sensing Image Retrieval Using Geospatial Foundation Models. arXiv 2024.

[5] Heo et al. Masking Augmentation for Supervised Learning. arXiv 2024.

[6] Wagstaff et al. Mars Image Content Classification: Three Years of NASA Deployment and Recent Advances. arXiv 2021.