Towards A Unified Deep Learning Architecture for Extraterrestrial Surface Perception

Brian Wu
Department of Computer Science, Stanford University
450 Jane Stanford Way, Stanford, CA 94305
brianwu@stanford.edu

Abstract

Autonomous robots exploring extraterrestrial surfaces require robust perception systems capable of semantic segmentation and depth estimation with limited computational overhead. We present a unified multi-task architecture using a SAM-based backbone with specialized prediction heads for jointly performing semantic segmentation and monocular depth perception on lunar terrain. This is done through using a multi-task objective function to balance segmentation accuracy with depth precision. In particular, we incorporate knowledge distillation from a pre-trained DPT model to enhance depth prediction capabilities. Our model is trained and evaluated on LuSNAR, a synthetic lunar perception dataset consisting of RGB images, depth maps, and segmentation labels. We achieve competitive segmentation performance with an mIOU of 0.7766, beating out CNN-based methods such as UNet++. We also achieve strong depth perception performance with an absolute relative error of 0.074 and an RMSE of 0.024 – outperforming Depth Anything v2 (current SOTA on general monocular depth estimation) on this particular task. We demonstrate that domain-specific, multi-task architectures effectively address the unique challenges of lunar perception tasks. Not only has this model learned to handle minimal color variation, extreme lighting conditions, and complex geological features, but it has also demonstrated the required computational efficiency necessary to support autonomous extraterrestrial exploration missions.

1. Introduction

Autonomous robots exploring and operating on the surface of extraterrestrial bodies such as the Moon and Mars require robust, power-efficient perception systems capable of safely navigating treacherous terrain with minimal human intervention. A strong level of autonomy for these scenarios is critical due to the time delay it takes to teleoperate spacecraft in these regions: for instance, one-way communication to a spacecraft on Mars could take as long as 21 minutes per transmission, which makes real-time remote control of these spacecraft extremely challenging and infeasible for complex mission scenarios with tight time-based constraints [1]. However, developing algorithms and models for autonomous rover-type spacecraft designed to operate on extraterrestrial surfaces comes with a unique set of challenges: modern perception-based foundation models are often trained on datasets containing a diverse set of objects; these models tend to perform well at disambiguating between different types and instances of objects. On the Lunar and Martian surfaces, there is a relatively small number of object classes, such as the sky, regolith (soil), rocks, craters, etc. The main challenges faced by perception algorithms in this domain relate to minimal color variation between objects, extreme lighting conditions (either high levels of glare from sunlight or large shadows that occlude objects), as well as complex geological features (rocks and craters generally do not have a uniform shape). This presents a complex scenario for any autonomous navigation stack.

While there are many types of tasks relevant to autonomous navigation on extraterrestrial surfaces, we consider two of the most paramount and difficult tasks in this domain for perception-based models: semantic segmentation and depth perception. In the semantic segmentation task, we take as input an RGB image tensor and assign a class label to every pixel in the image. Each of these class labels corresponds to an object category, and these labeled pixels are subsequently used to generate a segmentation map, which is a tensor with the same dimensions as the input RGB image but each pixel is colored according to its class label. This makes it easy to visualize the different types of objects that have been detected in the image. Semantic Segmentation can be performed using either Convolutional Neural Networks or Transformer-based models; an RGB image is used as input and a segmentation map of the same spatial dimensions is generated. In the depth per-

ception task, we take as input an RGB image tensor and compute a scalar value for every pixel in the image that indicates distance to the location where the image was taken. These scalar values are used to generate a depth map, which is a tensor with the same dimensions as the input RGB image but each pixel corresponds to the scalar depth/distance value at that location. Color maps can be used to visualize depth maps, with different colors in a gradient indicating the distance from the object to the camera. Like Semantic Segmentation, Depth Perception can be performed using either Convolutional Neural Networks or Transformer-based models (although we only consider the latter in this research); an RGB image is used as input and a depth map of the same spatial dimensions is generated.

This work was recently inspired by our participation in the 2025 edition of NASA's Lunar Autonomy Challenge (LAC) as part of Stanford's Navigation and Autonomous Vehicles Laboratory. The LAC involves using a digital twin of NASA's IPEx rover to explore and map the lunar surface [3]. Our solution for the competition involved off-the-shelf models for both of these tasks – namely, Depth Anything v2 for Depth Perception and LangSAM for Semantic Segmentation. We observed significant inefficiencies in traditional perception stacks such as the one used in the LAC; running separate models for both tasks involved lengthy query/inference times and higher compute requirements, thus reducing our observed mapping efficiency. Given the power and processing constraints of space-rated hardware, coupled with the aforementioned communication delays, we aim to explore in this research the development of a unified model architecture and evaluate how it performs on each of the aforementioned tasks with respect to dedicated semantic segmentation/depth perception models. As such, this work will address an existing gap in Computer Vision tasks for space-related applications while advancing resource-constrained perception systems for autonomous mobile robots.

2. Related Work

The jointly-trained unified perception model architecture in this work is primarily inspired by [14], which describes how depth perception capabilities can be integrated within SAM-type models. The authors accomplish this by incorporating what they term as the Prompt Deeper (which integrates RGB and depth features through knowledge distillation and bias correction and subsequently uses depth data to refine erroneous RGB features and generate depth-aware prompts for SAM) and Finer (which enhances segmentation accuracy by recovering missed camouflaged regions via mask reversion, self-filtering, and depth-guided self-attention operations) modules. We find their knowledge distillation approach relevant and incorporate a similar though simpler technique in our work.

[10] presents a multi-view stereo technique that allows SAM, a powerful pre-trained general-purpose semantic segmentation model capable of adapting to a variety of image features (that is SOTA on a variety of segmentation tasks), can be modified to outperform competing simple-task models by leverage multiple camera views [4], [9]. Although our approach focuses primarily on single-image perception, we incorporates part of the SAM modification strategy and prediction head construction into our single-view context; a similar heuristic is also discussed in [14] as well.

We use both CNN and Transformer-based baselines for the semantic segmentation task. This is because CNN-based models can run better within a resource constrained environment at the expense of accuracy and model capacity. [6] proves that CNN models can be jointly trained on segmentation and depth tasks using asymmetric annotations, and shows how multi-task loss functions (similar to the one implemented in this work) can be designed with competing objectives in mind. We incorporate these findings on loss balancing strategies with a more sophisticated backbone network from [4]. We choose UNet++ as our CNN baseline, which performs robustly against a variety of CNN segmentation benchmarks [15]. Furthermore, since we utilize a SAM-based backbone in our unified model, we evaluate performance on the semantic segmentation task against two additional Transformer-based models designed specifically with this task in mind. MaskFormer combines semantic, instance, and panoptic segmentation by generating mask predictions from global category queries and is relatively lightweight to run [2]. SegFormer uses an encoder-decoder architecture with a hierarchical transformer encoder that is coupled to a lightweight decoder [11]. As such, SegFormer is an excellent model for efficiently conducting the pixel-level semantic segmentation task, whereas SAM-family models were designed with open-world generalization in mind. For resource-constrained applications that requires precise class-specific segmentation (without necessarily using a highly general model such as SAM), SegFormer is a good choice and performs at a near-SOTA level on these tasks. We use fine-tuned versions of SegFormer, MaskFormer, and UNet++ as our baselines on the semantic segmentation task.

For the depth perception task, we consider the Depth Anything series of models. The original Depth Anything model, based on the Transformer architecture, used a mix of real-labeled and pseudo-labeled images combined with auxiliary supervision for robust performance on the monocular depth estimation task [12]. Depth Anything v2 improves the model by shifting to a synthetic dataset for supervised learning and introduces a teacher-student knowledge distillation framework during training, along with a DINOv2-based backbone that provides stronger semantic features and fine-grained image features [13]. This model is SOTA on depth

perception accuracy, efficiency, and generalization; we use a fine-tuned version of this model as our depth perception task baseline. A competitor to the Depth Anything series of models is the Intel DPT (Dense Prediction Transformer) architecture, which uses Vision Transformers as a feature extractor along with a decoder that processes multi-scale image features into a depth map [8]. DPT-type models have been able to achieve SOTA in some dense prediction benchmarks. We use DPT as the teacher for the depth prediction head due to it having less computational cost on average compared to Depth Anything v2; in particular, our experiments will determine if our SAM backbone in the jointly trained unified model can match Depth Anything v2's performance on the depth perception task without using a DINOv2 backbone to extract semantic image features.

3. Dataset

We incorporate the LuSNAR dataset in our research. LuSNAR is a comprehensive dataset designed for a suite of lunar surface perception tasks. It provides high-fidelity simulated lunar scenes, with each scene consisting of RGB images in PNG format, depth maps in PFM format, semantic segmentation maps with labeled pixels in PNG format, and LiDAR point clouds [5]. Constructed as an entirely synthetic dataset using Unreal Engine, LuSNAR provides 9 high-fidelity surface scenes with varying topographic relief and object density. This enables simulation of conditions ranging from simple flat terrain to treacherous mountainous terrain, along with a varying distribution of rocks and impact craters throughout. Throughout these scenes, LuSNAR is comprised of 13,006 sequences, with each sequence containing between 1,000 to 2,000 individual frames. A frame within a sequence consists of an RGB image in PNG format (42 GB total in the dataset) with a 1024×1024 resolution, paired with the following:

- Semantic Segmentation Map (356 MB of labels): 1024×1024 map with each pixel colored based on its class label. Valid class labels are $\{$lunar regolith, rocks, impact craters, mountains, sky$\}$.
- Depth Map (50 GB total): 1024×1024 map in 16-bit PFM format with each pixel position containing a scalar value that represents the distance to the camera.
- 3D Point Clouds: Approximately 10,000 points per frame with semantic labels for each point indicating if that point is regolith or part of a rock/crater.
- IMU data and ground truth poses.

We make use of the RGB images, semantic segmentation maps, and depth maps from each sequence. The 3D Point Clouds and IMU poses are relevant for SLAM applications, which we do not consider in this work. We follow a suggested 80-20 split for training and validation data respectively which translates to just over 10,000 sequences for the training set and just over 2,000 sequences for the validation set. Within this dataset, we index primarily on geometric properties as the main feature since there is little color variation between objects (except for the sky). The dimensions of the rocks and craters are accurate based on knowledge about the lunar surface; furthermore, we also focus on the relative scale between objects. For instance, a mountain is going to appear far larger and more distant compared to a rock, despite both having similar colors and shapes in certain cases.

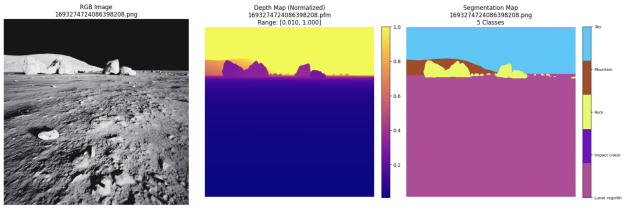

Figure 1. A dataset example from LuSNAR. (Left) RGB image of the lunar surface. (Center) Corresponding depth map. (Right) Corresponding segmentation map with labels applied.

To preprocess and augment the dataset, we first created a pipeline to normalize the RGB images with an ImageNet mean/standard deviation. We also apply normalization over the depth maps as well before feeding them into any model. To augment the data, we follow standard practice by introducing random horizontal flipping, rotations of up to 10° in either direction, as well as brightness/contrast adjustments to mimic the extreme lighting conditions often found on the Lunar surface.

4. Technical Approach and Experiments

4.1. Baseline Methods

We use single-task models as baselines for the semantic segmentation and depth perception tasks to evaluate our unified model against. On the semantic segmentation task, our baselines are SegFormer [11], MaskFormer [2], and UNet++ [15]. On the depth perception task, we utilize Depth Anything v2 [13]. All of these models are fine-tuned on LuSNAR; we discuss the fine-tuning recipes for each of these models in section 4.4.

SegFormer is a Transformer-based model that performs semantic segmentation via a hierarchical Transformer encoder with a lightweight MLP-based decoder. The encoder is used to extract multi-scale features, and the decoder is used to fuse these features together and subsequently project them into a depth map. Given an input RGB image $X \in \Re^{H \times W \times 3}$, the encoder produces feature maps at 4

different scales

$$F_i \in \Re^{\frac{H}{2^i} \times \frac{W}{2^i} \times C_i} \; \forall i \in \{1, 2, 3, 4\}$$

using self attention. The decoder first takes the features and passes them through an MLP to unify the channel dimension; subsequently, these features are upsampled and concatenated together. The remaining two MLP layers, respectively, fuse the concatenated features and predict the segmentation map with a resolution of $\frac{H}{4} \times \frac{W}{4} \times N_{cls}$ where N_{cls} represents the number of semantic categories an object could belong to [11].

The MaskFormer model reformulates the semantic segmentation task as a mask prediction task. Here, the model uses bipartite patching to align predictions with ground truth and output a fixed set of mask-class pairs. Given an input RGB image $X \in \Re^{H \times W \times 3}$, a pixel decoder produces per-pixel embeddings $E \in \Re^{H \times W \times d}$, and a transformer decoder outputs N mask embeddings $M \in R^{N \times D}$. The mask prediction step is performed via a dot product of the mask embeddings and pixel embeddings $m_i = \sigma(M_i E^T)$, and then classification is accomplished by applying a linear layer on the per-segment embeddings followed by a Softmax classifier to obtain the probability predictions [2].

UNet++ is a Convolutional Neural Network following an encoder-decoder architecture with nested skip pathways that aim to bridge the semantic gap between encoder and decoder feature maps. Compared to a vanilla U-Net, this model has convolution layers on skip pathways (along with incorporating dense skip connections on these pathways), and uses deep supervision. Given an input RGB image $X \in \Re^{H \times W \times 3}$, the nested skip connections are computed via

$$x^{i,j} = \begin{cases} \mathcal{H}\left(x^{i-1,j}\right), & j = 0 \\ \mathcal{H}\left(\left[\left[x^{i,k}\right]_{k=0}^{j-1}, \mathcal{U}(x^{i+1,j-1})\right]\right), & j > 0 \end{cases}$$

where $\mathcal{H}$ represents a convolution operation followed by an activation function, $\mathcal{U}$ is an upsampling layer, and $X^{i,j}$ is the feature at the down-sampling encoder layer i, dense block convolution layer j [15]. Additionally, Deep Supervision, where the loss is computed at multiple decoder depths, enables better gradient flow and improved feature fusion by allowing the model to balance accuracy versus speed.

Depth Anything v2 is based on the Vision Transformer (ViT) architecture and is adapted for the monocular depth estimation task. As it is pretrained on a large-scale depth dataset, it is designed to be a general-purpose model for a variety of depth perception scenarios. Given an input RGB image $X \in \Re^{H \times W \times 3}$, the ViT backbone extracts features at multiple scales, and then the DPT-based decoder (which itself uses features from DINOv2) works by predicting a depth map $\hat{D}_i$ from the ViT-extracted features F_i. This architecture is paired with two loss functions: a scale- and shift-invariant loss as well as a gradient matching loss, which enables it to robustly estimate depth values from a single image [13].

4.2. Unified Multi-Task Architecture

We propose a unified multi-task architecture for semantic segmentation and depth perception tasks. We begin with a frozen vision encoder from SAM-ViT-Huge, which serves as a backbone feature extractor for both the segmentation and depth estimation tasks. Then, for each task, we create a task-specific prediction head. Both of these heads were designed initially to each consist of a convolutional layer with a 3×3 filter, 128 channels, batch normalization, and a ReLU activation function, along with a 1×1 convolutional layer that either maps to 5 class outputs (in the case of the segmentation prediction head), or a single channel scalar-valued output (in the case of the depth estimation prediction head). Sequentially, we represent this architecture as follows:

$$F = \Phi(I)$$

where Φ is the SAM-ViT-Huge vision encoder and $F \in \Re^{B \times C \times H' \times W}$ represents the extracted features;

$$S = \Psi_{seg}(F)$$

where Ψ_{seg} is the segmentation head, yielding logits for each of C classes, and

$$D = \Psi_{depth}(F)$$

where Ψ_{depth} is the depth perception head which yields a scalar-valued output.

This architecture has evolved over our experimentation procedure. First, we unfreeze the last 4 blocks of the SAM-ViT-Huge encoder as the images from LuSNAR might be significantly different from the examples in the dataset that SAM was trained on. While the lightweight prediction heads worked well for the segmentation task (likely owing to the SAM-based backbone), empirical results indicated that such a design underperformed on the depth perception task. As such, we draw inspiration from DPT-based models and create a new depth prediction head incorporating dilated convolutions for enlarging receptive fields without resolution loss as well as an attention mechanism (which emphasizes structurally salient features), both of which should help the model capture global depth structure and fine-grained edge and smoothness details [8]. To further enhance the performance on the depth prediction task, we introduce a Knowledge Distillation (KD) process by using a pre-trained DPT-Large model as a teacher model, while the prediction head in our unified model serves as the student model. KD was introduced to help the student model learn richer representations even when direct ground-truth supervision via LuSNAR is sparse or noisy. We use structural matching by aligning the gradient maps of the student and

teacher model outputs to better preserve edge contours and environment topography. Furthermore, we also normalize the teacher model's output per sample to match the student model's distribution. Therefore, our student model should be able to learn from annotated ground-truth depth maps in LuSNAR, but also mimic structural qualities found in the pre-trained teacher's output.

Given a student model S and a pretrained teacher model T, we wish to transfer structural knowledge from the teacher model's depth predictions $D_T \in \Re^{B\times1\times H\times W}$ to the student model's depth predictions D_S. We apply per-sample normalization to match the dynamic range and distribution of the student predictions as follows:

$$\mu_T = \text{mean}(D_T),\ \sigma_T = \text{std}(D_T)$$

$$\mu_S = \text{mean}(D_S),\ \sigma_S = \text{std}(D_S)$$

$$\tilde{D}_T = \frac{D_T - \mu_T}{\sigma_T + \epsilon}\sigma_S + \mu_S$$

where $\tilde{D}_T$ is the normalized teacher depth map and ϵ is a small positive constant to help maintain numerical stability.

Figure 2 describes the overall unified model architecture with DPT-based Knowledge Distillation on the depth prediction head.

4.3. Multi-Task Objective

We propose training the unified model described in section 4.2 on both segmentation and depth perception tasks simultaneously using a multi-task loss function given by

$$\mathcal{L}_{total} = \lambda_{seg} * \mathcal{L}_{seg} + \lambda_{depth} * \mathcal{L}_{depth} + \lambda_{KD} * \mathcal{L}_{KD}$$

where $\mathcal{L}_{seg}$ is the cross-entropy loss for the segmentation task, given by

$$\mathcal{L}_{seg} = -\sum_{c=1}^{C} y_c \log(\hat{y}_c)$$

with C indicating the total number of classes, and $\mathcal{L}_{depth}$ is the $L1$ loss for the depth estimation task given by

$$\mathcal{L}_{depth} = \frac{1}{N}\sum_{i=1}^{N} |d_i - \hat{d}_i|$$

Furthermore, to better capture structural detail such as terrain edges, we define a gradient-based distillation loss as follows:

$$L_{grad} = ||G_x(D_S) - G_x(\tilde{D}_T)||_1 + ||G_y(D_S) - G_y(\tilde{D}_T)||_1$$

where G_x and G_y denote convolutions with horizontal and vertical Sobel kernels (respectively). Furthermore, we also use a pixel-wise $L2$ loss to align the overall depth structure:

$$\mathcal{L}_{pixel-wise} = ||D_S - \tilde{D}_T||_2^2$$

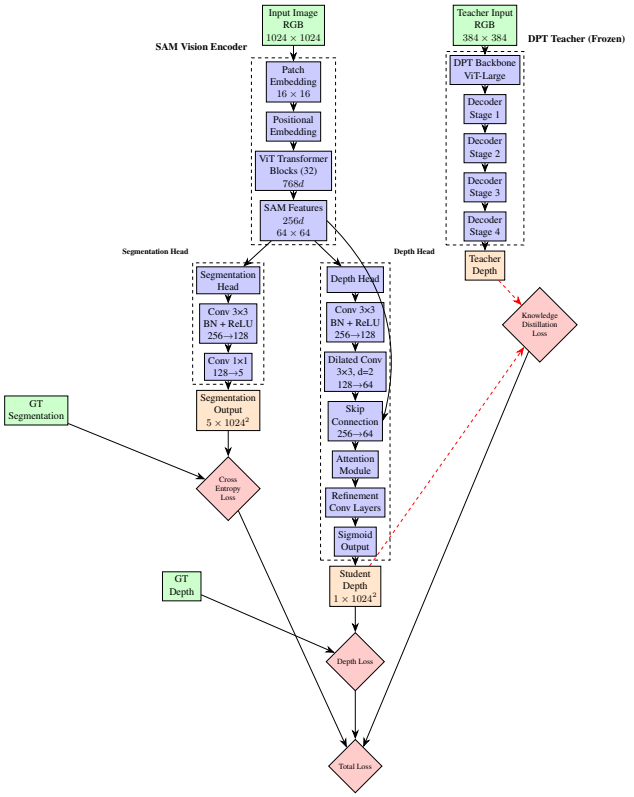

Figure 2. Unified Model Architecture with Knowledge Distillation. The unified model uses a SAM Vision Encoder as the backbone with specialized heads for segmentation and depth estimation (student model). The frozen DPT teacher provides depth knowledge distillation.

We combine the gradient loss term with the alignment loss term to obtain

$$L_{KD} = \alpha * \mathcal{L}_{grad} + (1 - \alpha) * \mathcal{L}_{pixel-wise}$$

where $\alpha \in [0, 1]$ is a weighting parameter for the KD loss.

We include $\lambda_{seg}, \lambda_{depth}, \lambda_{KD}$ as task weights that collectively weigh the relative importance of each of these objectives in the final loss formulation. For the purpose of our experiments, we set $\lambda_{seg} = 1.0$, $\lambda_{depth} = 5.0$, and $\lambda_{KD} = 0.05$. We upsample predictions to match ground-truth dimensions using bilinear interpolation before computing the losses to be robust to different output resolutions.

4.4. Training Strategy and Hyperparameter Tuning

On the processed LuSNAR dataset described in section 3, we fine-tune all of our baseline models using a similar recipe consisting of 5 epochs, as well as the Adam Optimizer (PyTorch implementation) for all baseline models. We also found that using AdamW with a standard weight decay parameter of 0.01 led to more stable fine-tuning for the Depth Anything v2 and MaskFormer models. For the learning rate, we manually tuned within $[1*10^{-3}, 5*10^{-5}]$.

We found optimal learning rates of $5 * 10^{-5}$ for Depth Anything v2, $1 * 10^{-5}$ for MaskFormer, $5 * 10^{-5}$ for SegFormer, and $1 * 10^{-4}$ for UNet++. We follow a similar training recipe for the unified model, except the number of epochs was increased to 10 from 5 because we trained two prediction heads jointly and the segmentation prediction head was not initialized from any pre-trained weights. We use the Adam optimizer and find that a weight decay parameter of $1 * 10^{-5}$ and a learning rate of $1 * 10^{-4}$ were optimal for a smooth and stable training process. In our experiments, given the large image sizes and teacher model inference cost, we use a small batch size of 4 to reduce memory load on our setup.

5. Experiments and Results

5.1. Experiment Design

All models were fine-tuned via the HuggingFace and PyTorch libraries; for all models except UNet++ described above, we use the standard implementation of the model as according to their respective papers and HuggingFace. UNet++'s implementation is as according to its original paper and the *segmentation-models-pytorch* library implementation. We then constructed a dataset pipeline for LuSNAR as well as a training and evaluation pipeline for each of these model implementations.

For the semantic segmentation task, we evaluate on the Mean Intersection Over Union (mIOU) metric, which is the overlap between segmented objects in the predicted segmentation map versus the ground truth segmentation map divided by the total area of the image covered by the union of the two. For the depth estimation task, we use the absolute relative error (ARE) and root-mean-square error (RMSE) between pixel values of the ground truth depth map versus the predicted depth map. We then compute the $\delta 1$, $\delta 2$, and $\delta 3$ values, which are the percentage of pixels where the ratio between the prediction and ground truth is less than 1.25, 1.25^2, and 1.25^3 (respectively).

We first fine-tuned all of the baseline models using the training recipes outlined in section 4.4 and obtained baseline results on standard metrics for the semantic segmentation and depth perception tasks, respectively. For the unified model, we begin with the two basic CNN-based prediction heads as outlined in section 4.2. Each version of this model was trained on the same recipe described in section 4.4. This initial version of the model performed on par with UNet++ on the segmentation task, but exhibited significantly worse performance on the depth perception task, with an absolute relative error twice that of Depth Anything v2 and a $\delta 1$ value of only about 21%. Subsequently, we added dilated convolutions, an attention block, and a skip connection to the depth perception head, but this still did not lead to significant performance gains on the depth perception metrics. Finally, we unfroze the last 4 blocks of the SAM encoder and introduced knowledge distillation to the depth prediction head using a pre-trained DPT model. This was able to improve depth perception performance significantly while beating UNet++ performance slightly. All fine-tuning and training experiments were conducted on a Linux server with a single NVIDIA GeForce RTX 3090 containing 24 GB of memory managed by the author [7].

5.2. Results and Evaluation

In table 1, we report the overall results on the mIOU metric for the semantic segmentation task of the unified model against the segmentation baselines. In table 2, we report the overall results on the absolute relative error, RMSE, and $\delta 1$, $\delta 2$, $\delta 3$ metrics for the depth perception task of the unified model against Depth Anything v2.

Table 1. Semantic Segmentation Performance Comparison

Model	**Mean IoU**
UNet++	0.7681
MaskFormer	0.8955
SegFormer	0.9686
Unified	0.7766

Table 2. Depth Estimation Performance Comparison

Model	**ARE**	**RMSE**	$\boldsymbol{\delta_1}$	$\boldsymbol{\delta_2}$	$\boldsymbol{\delta_3}$
DA v2	0.688	0.080	0.901	0.954	0.972
Unified	0.074	0.024	0.947	0.986	0.997

Subsequently, in figure 3 we display the segmentation and depth maps generated by the unified model and compare to the to the ground truth segmentation and depth maps from LuSNAR. We also display depth maps generated by the teacher model as well as the depth difference from the Knowledge Distillation process.

We also compute loss curves for the unified model and for each of the baselines, along with qualitative visualizations of segmentation and depth maps generated by the baseline models. These results are included in the appendix for the sake of readability of this paper.

5.3. Discussion

We observe that our unified multi-task model performs competitively across both the segmentation and depth estimation metrics. We achieve a mean IoU of 0.7766 on the semantic segmentation task, and we also demonstrate strong depth estimation metrics with an Absolute Relative Error of 0.074, an RMSE of 0.024, and values above 94% for $\delta 1$, $\delta 2$, and $\delta 3$. These metrics outperform or match the performance of existing general-purpose segmentation and depth perception models on lunar surface environments, and thus

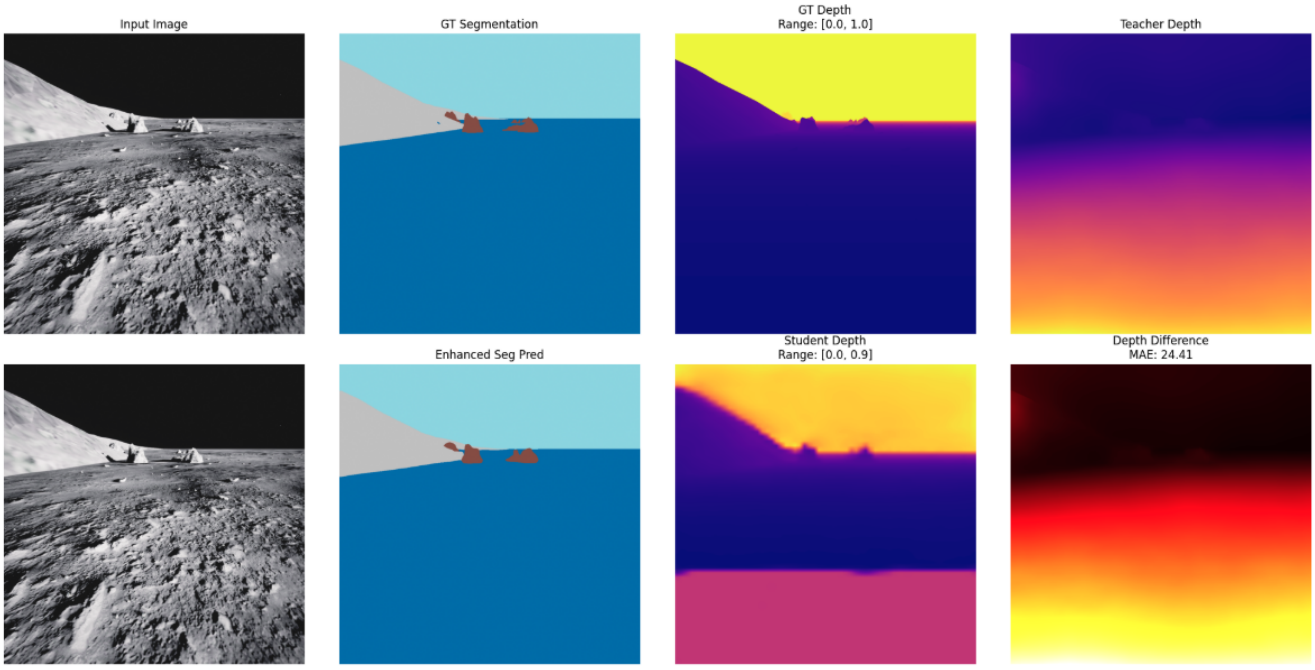

Figure 3. Ground Truth Segmentation Maps/Depth Maps and Unified Model-predicted Segmentation Maps and Depth Maps. (Top Row, from left to right) Input RGB Image, Ground Truth Segmentation Map, Ground Truth Depth Map, Teacher Depth; (Bottom Row, from left to right) Input RGB Image, Predicted Segmentation Map, Predicted Depth Map, Depth Difference.

validates our architectural design choices along with task-specific optimizations for our unique scenario.

In our solution for the 2025 Lunar Autonomy Challenge, we strongly preferred UNet++ for segmentation due to its CNN-based architecture enabling significant reductions in computational cost. Given the extremely constrained compute environment we were in, having the most computationally efficient model was preferable to gaining a few extra points in efficiency. Our model outperforms UNet++ by a modest margin, and this is attributed to the powerful SAM backbone which provides robust feature representations that, in the case of a unified model, can match and exceed the performance of the complex CNN-based architecture from UNet++. In particular, we believe that SAM's ViT components, which are pretrained on massive segmentation datasets, are useful for high-fidelity feature extraction within complex scene understanding tasks. Furthermore, jointly optimizing segmentation and depth estimation in this regard provides some implicit regularization and feature sharing benefits; we believe that the depth estimation branch of the model can provide some geometric understanding using the depth values that may help the segmentation prediction head inform some of its segmentation decisions, such as distinguishing between semantically similar but geometrically distinct regions. At the same time, our model underperforms the two Transformer-based metrics; MaskFormer achieved an mIOU of 0.8955 and SegFormer achieved the best mIOU value of 0.9686 in our experiments. We attribute this to the design of our segmentation perception head: given real-life compute requirements in our scenario, we opted to choose a CNN-based architecture. While this architecture was lightweight and simple to implement by stacking it on top of the SAM feature extractor backbone, MaskFormer and SegFormer both leverage Transformer-based architectures end-to-end, enabling more sophisticated attention mechanisms and global context modeling (along with higher model capacity) that make them particularly good at generalizing to the lunar terrain segmentation task after fine tuning. In particular, MaskFormer uses the mask classification paradigm with learnable queries, which might excel at handle overlapping and hierarchical semantic regions such as rocks sitting on top of lunar regolith. On the other hand, SegFormer uses a hierarchical transformer architecture, which helps it better capture multi-scale feature interactions. Our CNN prediction head is likely unable to achieve the same level of global context integration and struggles with complex spatial relationships between objects. Furthermore, the Transformer-based models benefit from more sophisticated data augmentation and training recipes tailored to their architecture, while we prioritize the multi-task learning objective which may not have fully optimized segmentation performance as it focused more on improving the Depth Estimation task performance.

On the Depth Estimation task, our model has achieved

better performance than Depth Anything v2 in terms of both the error metrics and the $\delta 1$, $\delta 2$, $\delta 3$ metrics. Depth Anything v2 is designed as a general purpose monocular depth estimation model and is trained on lots of diverse imagery. Even with the fine tuning introduced in our work, we find that our unified model can perform better. We designed the unified model specifically with lunar terrain characteristics in mind, such as the unique lighting conditions and surface textures across different types of objects. These results lead us to believe that there exists a domain gap between the terrestrial training data within Depth Anything v2 and the surface conditions likely to be encountered by a robot operating on the lunar surface. This prevented Depth Anything v2 from performing better on the lunar terrain depth estimation task. In retrospect, we would have unfrozen more layers of the Depth Anything v2 model; we opted not to do so due to training hardware limitations. Our results suggest that the features extracted from the SAM backbone, when combined with knowledge distillation applied to the prediction head, were beneficial to the unified model performing well on the Depth Estimation task. Using a distilled prediction head on this task allowed us to modify our architecture with dilated convolutions (which resulted in expanded receptive fields) as well as refining our depth features using attention mechanisms. The relationships between different objects as determined by the SAM backbone likely presents a good starting point for estimating depth based on where the boundaries between these objects lie. One other observation is that the outputs of both our model and Depth Anything v2 exhibit some blur in the predictions compared to the ground truth data. This is likely due to limitations with monocular depth: there exists a scale ambiguity where multiple depth configurations can produce the same image projection without a stereo image to verify against. The lunar terrain environment likely suffers from this due to the lack of color and object type variation across scenes. It seems that both models are using image-specific features such as the texture gradients to infer depth; this will not work as well for lunar terrain scenes due to the relatively uniform texture (in terms of color) on the lunar surface. As such, the visual cues used by the model to estimate depth are not robust enough, which leads to the slight blur as seen in the depth predictions. Furthermore, we observe a pink region at the bottom of the plotted student depth; after examining the values in this region of the image, we find that the predicted values are very similar to the ground truth depth (which does not affect the quality of the prediction during evaluation). We attribute this artifact to a plotting error.

6. Conclusion

We demonstrate the viability of joint multi-task learning for two challenging lunar perception tasks with a unified perception model capable of performing semantic segmentation and depth estimation on the lunar surface. Compared against established baselines in both tasks, our SAM-based architecture achieved competitive results with an mIOU of 0.7766 for segmentation and strong depth perception performance with an ARE of 0.074 and an RMSE of 0.024. While our model was able to outperform Depth Anything v2 and UNet++, it still has potential for improvement in the segmentation task as the two transformer-based segmentation baselines were able to achieve higher mIOU values. SegFormer's end-to-end hierarchical transformer architecture, along with MaskFormer's query-based mask classification approach, enable them to handle long-range dependencies, multi-scale features, and overlapping semantic regions well; all of these exist within our lunar terrain environment. Our CNN-based prediction head, even when augmented with features extracted from SAM, is still unable to match the performance of these Transformer-based models; however, it was able to slightly edge out the performance of an advanced CNN-based architecture (UNet++). We attribute our strong performance on the depth estimation task to our architecture, combining SAM-extracted features, a transformer-based prediction head using the DPT architecture, as well as knowledge distillation from a pretrained DPT model. This architectural design is critical towards closing a domain gap unique to our task characterized by a lack of color variation and harsh lighting conditions. In our joint multi-task learning approach, we have developed a computationally efficient model that can be widely deployed across autonomous robots operating in resource-constrained scenarios on extraterrestrial surfaces. These types of unified, multi-task perception models will become compelling choices for navigation stacks running onboard autonomous extraterrestrial surface exploration missions.

6.1. Future Work

Assuming we have more available compute, we will replace the CNN-based prediction head with a lightweight transformer-based architecture; the transformer attention is likely beneficial towards helping the model generalize better to segmentation features and interactions between objects (as has been shown with the segmentation baselines). Furthermore, given the lack of color variation on the lunar surface, we note that the edges and interactions between objects are critical towards performing well on segmentation, and we are looking into uncertainty estimation techniques to identify and handle ambiguous regions. On the depth estimation front, we are looking at modifying our objective function to handle sharp depth prediction, such as an L2-based loss (which may be useful as a sparse solution is not necessarily useful here) and adversarial training. Finally, taking inspiration from data adaptation recipes from large pre-trained models, we will introduce more domain-specific data augmentation strategies designed for lunar terrain.

7. Acknowledgments

We would like to thank the CS 231N Course Staff for their instruction on the topics covered by this research. Additionally, we would like to thank Professor Grace Gao and Adam Dai of the Stanford Navigation and Autonomous Vehicles Laboratory (NAV Lab). While this research did not make use of any NAV Lab resources and associated work was done independent of existing NAV Lab projects, we wish to thank Professor Gao and Adam for enabling the author's participation in the 2025 NASA Lunar Autonomy Challenge and providing extremely valuable technical guidance for this research.

8. Software Packages

All of the code for this research was written in Python 3.10.16. All software packages and their corresponding version numbers used in this research are displayed in table 3.

Table 3. Python packages and versions used in this work.

Python Package Name	**Version Number**
NumPy	1.26.4
Pandas	1.3.4
PyTorch	2.7.0
TorchVision	0.22.0
(HuggingFace) Transformers	4.51.3
(HuggingFace) Datasets	3.6.0
(HuggingFace) Evaluate	0.4.3
Matplotlib	3.10.1
scikit-learn	1.6.1
segmentation-models-pytorch	0.5.0
Pillow	11.2.1
Albumentations	2.0.6
Wandb	0.19.11

References

[1] Mars communications disruption and delay. Technical report, NASA, 2023. 2023 Moon to Mars Architecture Concept Review White Paper. 1

[2] B. Cheng, A. G. Schwing, and A. Kirillov. Per-pixel classification is not all you need for semantic segmentation, 2021. 2, 3, 4

[3] Johns Hopkins Applied Physics Laboratory. Lunar autonomy challenge. https://lunar-autonomy-challenge.jhuapl.edu/, 2024. Accessed: April 2025. 2

[4] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment anything, 2023. 2

[5] J. Liu, Q. Zhang, X. Wan, S. Zhang, Y. Tian, H. Han, Y. Zhao, B. Liu, Z. Zhao, and X. Luo. Lusnar: A lunar segmentation, navigation and reconstruction dataset based on multi-sensor for autonomous exploration. *arXiv preprint arXiv:2407.06512*, 2024. 3

[6] V. Nekrasov et al. Real-time joint semantic segmentation and depth estimation using asymmetric annotations. *arXiv preprint*, 2019. 2

[7] NVIDIA Corporation. Nvidia ampere ga102 gpu architecture whitepaper. Technical report, NVIDIA Corporation, 2020. Version 1.0. 6

[8] R. Ranftl, A. Bochkovskiy, and V. Koltun. Vision transformers for dense prediction, 2021. 3, 4

[9] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer. Sam 2: Segment anything in images and videos, 2024. 2

[10] M. Shvets et al. Joint depth prediction and semantic segmentation with multi-view sam. In *Proceedings of the Computer Vision Foundation*, 2023. 2

[11] E. Xie et al. Segformer: Simple and efficient design for semantic segmentation with transformers. *arXiv preprint*, 2021. 2, 3, 4

[12] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao. Depth anything: Unleashing the power of large-scale unlabeled data, 2024. 2

[13] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024. Accepted by NeurIPS 2024. Project page: https://arxiv.org/abs/2406.09414. 2, 3, 4

[14] Z. Yu et al. Exploring deeper! segment anything model with depth perception for camouflaged object detection. *arXiv preprint*, 2024. 2

[15] Z. Zhou et al. Unet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE Transactions on Medical Imaging*, 2019. 2, 3, 4

9. Appendix

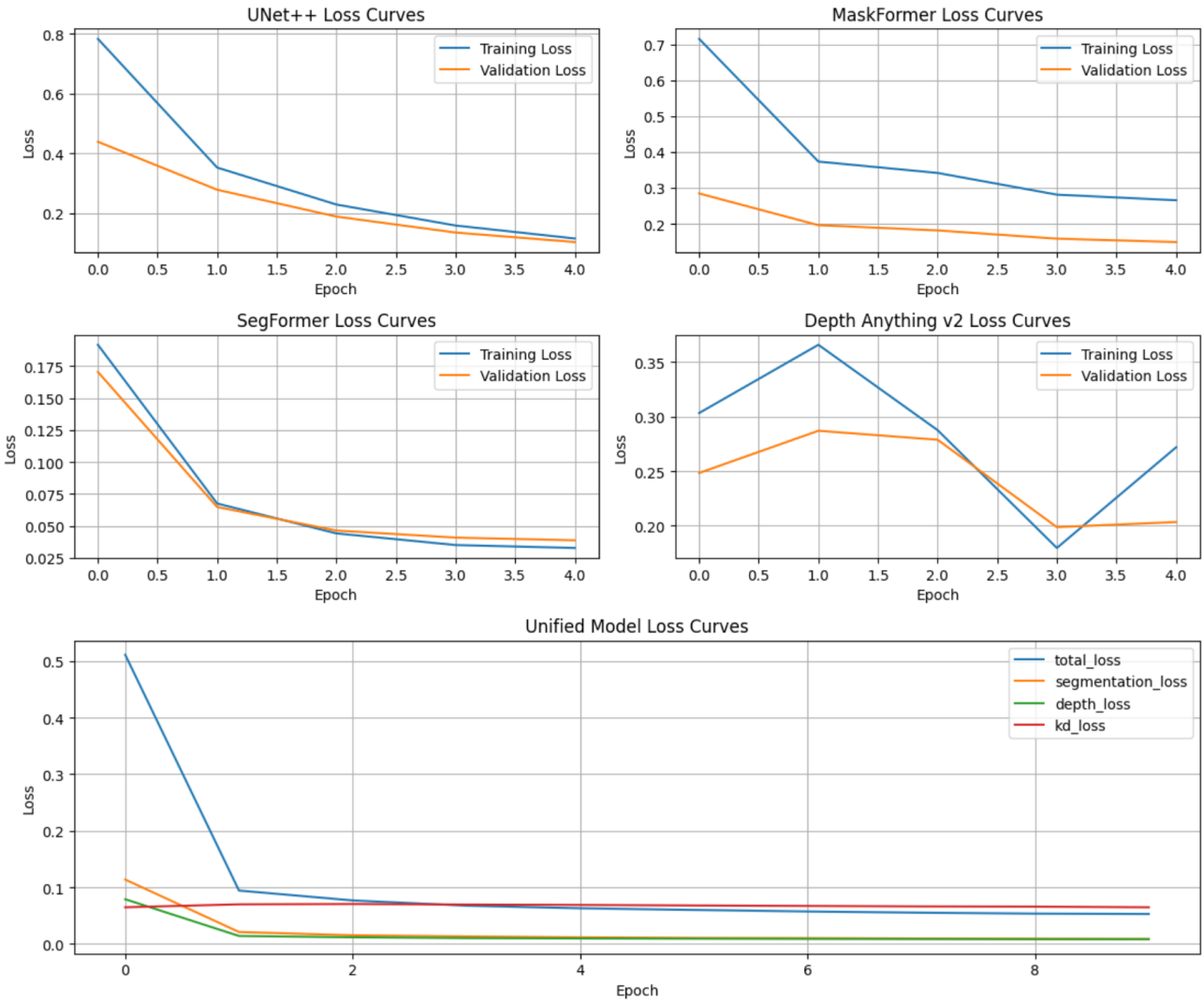

Figure 4. Training and Validation Loss Curves (per epoch) for baseline models and the unified model. (Top Left) UNet++; (Top Right) MaskFormer; (Center Left) SegFormer; (Center Right) Depth Anything v2; (Bottom) Unified Model.

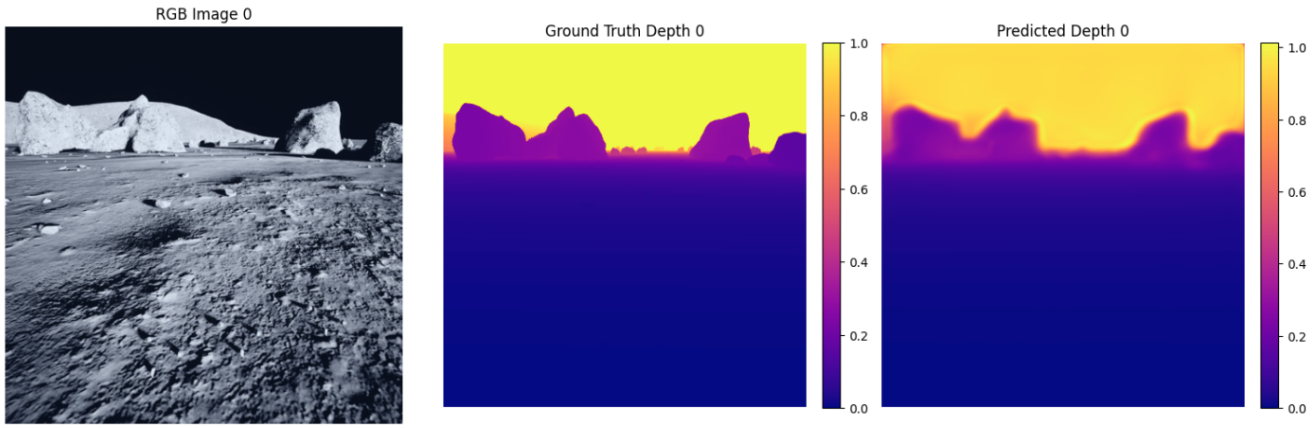

Figure 5. A sample of Depth Anything v2's Predictions on the Validation Set. (Left) Ground Truth RGB Image; (Center) Ground Truth Depth Map; (Right) Predicted Depth Map.

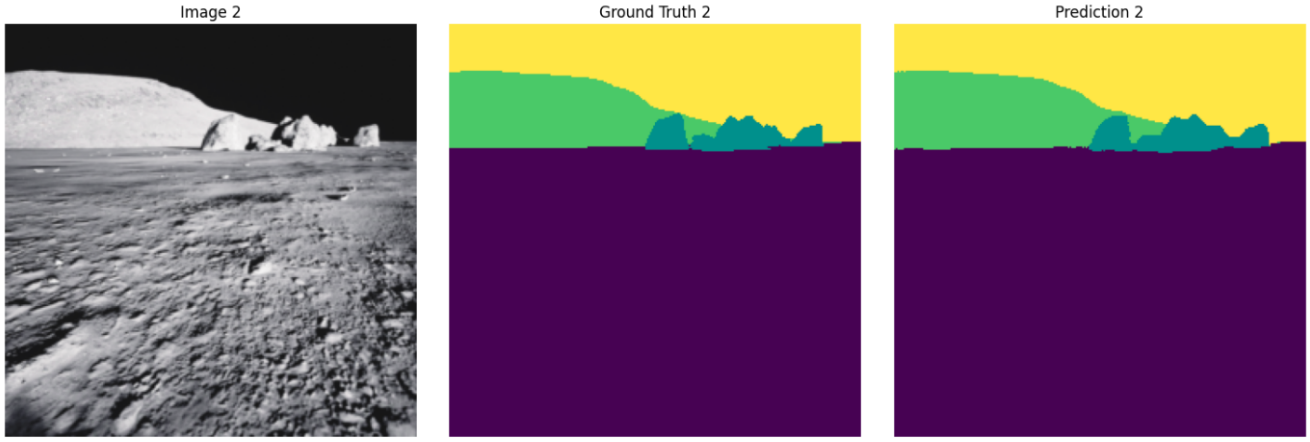

Figure 6. A sample of UNet++'s Predictions on the Validation Set. (Left) Ground Truth RGB Image; (Center) Ground Truth Segmentation Map; (Right) Predicted Segmentation Map.

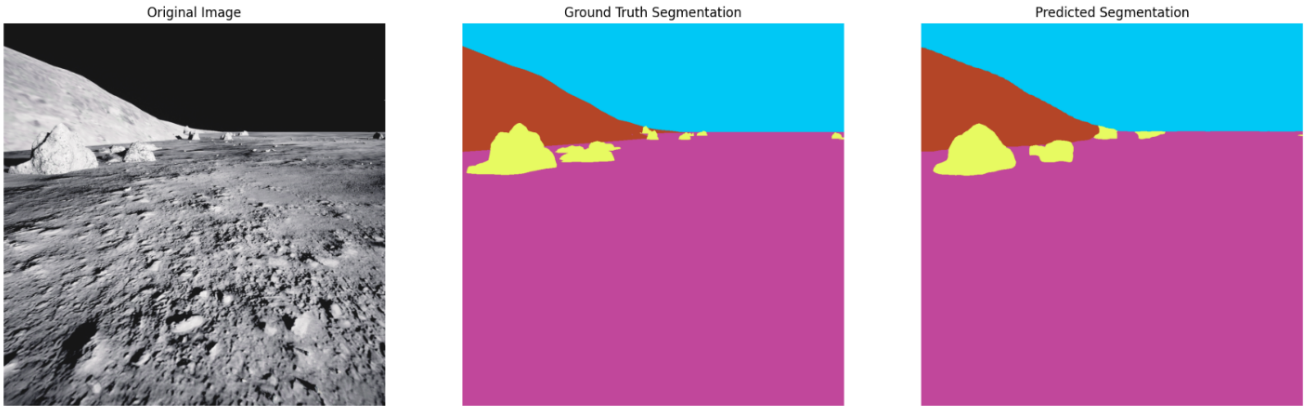

Figure 7. A sample of MaskFormer's Predictions on the Validation Set. (Left) Ground Truth RGB Image; (Center) Ground Truth Segmentation Map; (Right) Predicted Segmentation Map.

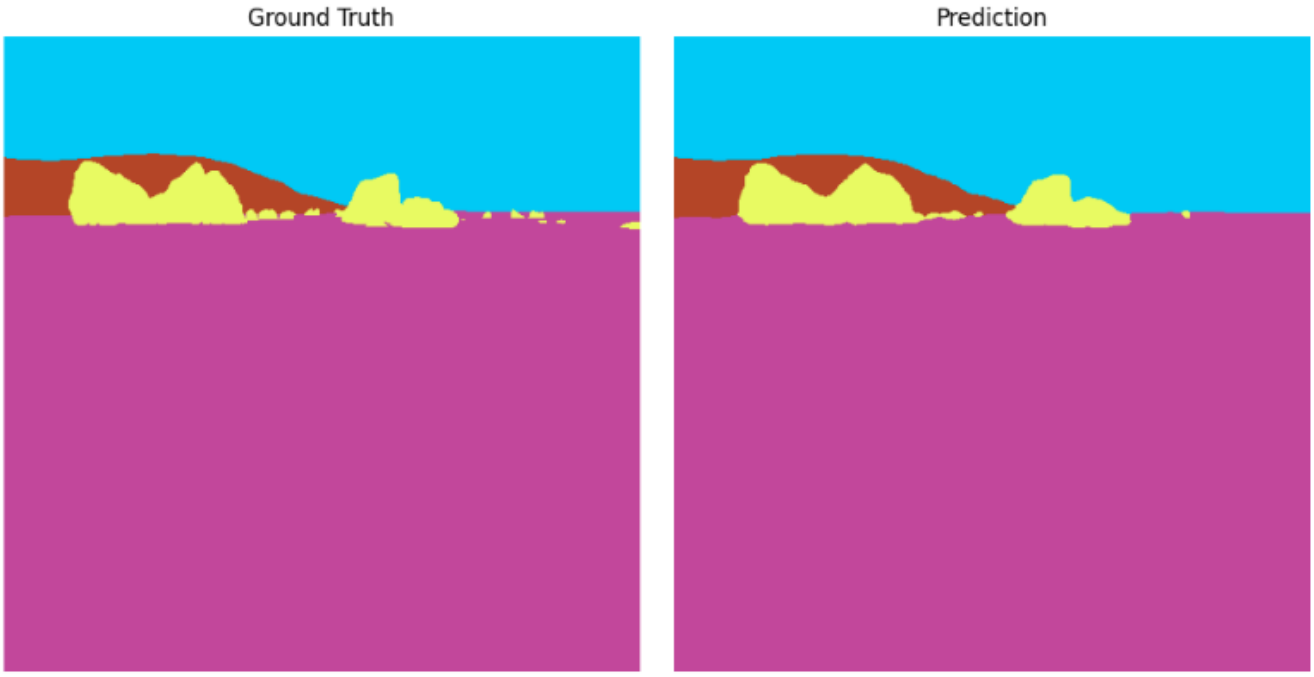

Figure 8. A sample of SegFormer's Predictions on the Validation Set. (Left) Ground Truth Segmentation Map; (Right) Predicted Segmentation Map.