RAS 598: Space Robotics and AI (Spring 2025)
March 27, p1~7
ASU ID: 1234000809

Semantic Segmentation of Lunar Surface

Kaito Namatame (1234000809)

Robotic and Autonomous Systems (Mechanical and Aerospace Engineering), Arizona State University

Abstract

Semantic segmentation is a important method to navigate robot or rover on unstructured and planetary environments such as the lunar surface. This project presents a neural network architecture for rover-based ground segmentation on the lunar surface. The architecture is processed with an U-net framework with a pre-trained CNN(Convolutional Neural Network), VGG16 as encoder and U-net decoder. The model is designed to classify lunar terrain features including rocks, sky and lunar surface. This project aims to contribute in lunar landscape segmentation using deep learning techniques.

1. Introduction

Semantic segmentation, which involves labeling each pixel in an image with a semantic category, is a powerful technique for unstructured and planetary environments. This allows the use of semantic information in the path planning.

In this project, I implemented an encoder-decoder neural network based on a U-net architecture with a VGG16 as an encoder, specifically developed for rover-based lunar terrain segmentation. The model which is trained on an artificial dataset of rover-based lunar landscape images, classifies terrain into four categories. By employing a pre-trained VGG16 model, frozen to utilize ImageNet weights, the architecture provides robust feature extraction to achieve accurate segmentation. This project explores lunar scene understanding through deep learning, evaluates the model's performance. U-net provides a high resolution image, of the same size as the input image, where each picture corresponds to a specific class.

2. Method

Scope of this project is to develop a robust semantic segmentation model. The implemented model is an encoder-decoder neural network based on the U-net framework, with VGG16 as the encoder. VGG16 is pre-trained model on ImageNet and is integrated with its top layers excluded and weights frozen to leverage its robust feature extraction capabilities.

RAS 598: Space Robotics and AI (Spring 2025)
March 27, p1~7
ASU ID: 1234000809

3. Dataset

A dataset with artificial rover-based images which depicts lunar landscapes was utilized for training and validation of the proposed architecture. The dataset was created by the Space Robotics Group of Keio University in Japan. It contains 9,766 artificial renders of rocky lunar landscapes, and their segmented masks (the 4 classes are the large rocks, small rocks, sky and ground.

Images are RGB renderings of lunar landscapes, while masks are grayscale images. The data is split with an 80:20 ratio for train and evaluation.

4. Model Architecture

U-net architecture is used for this semantic segmentation project. The U-shaped model of U-net is separated in two main components.

- First part is contraction path (encoder) that reduces the image dimensions, increasing the feature maps while learning to classify the desired features.
- Second part is the decoder which increasing the resolution of the output and decreasing the feature maps by Upsampling that replace pooling operations. To restore details of images, four skip connections from the encoder are utilized.
- To improve the performance of this segmentation model, the transfer learning is utilized, using pre-trained VGG16 as an encoder components of the U-net, since the U-net includes 23,750,180 trainable parameters.

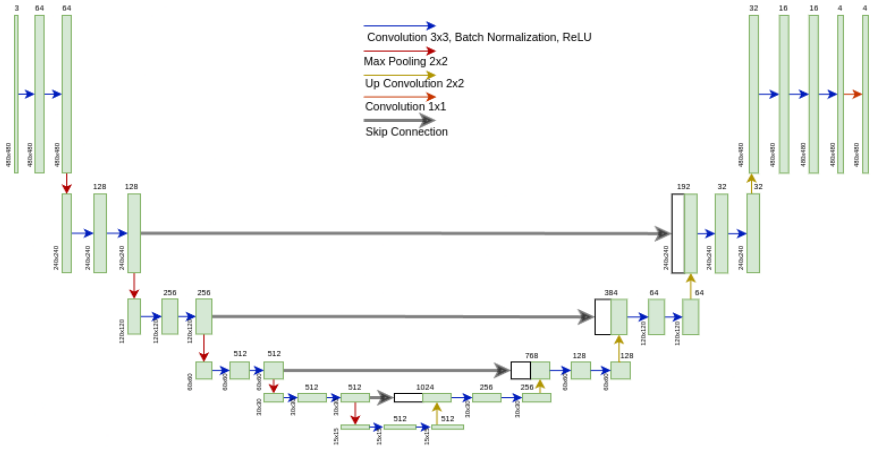


Figure 1: U-net is composed by convolutional, BatchNormalization and ReLU layers. This procedure is repeated and applied in every single pixel of an image.The encoder downsamples the image through the MaxPooling layer, and the decoder upsamples the images using the Up Convolution layers. The Concatenate layer generates the skip connections between the encoder and decoder components. At the layer, Softmax, which is the activation function is utilized to export the segmentation map for each input image.

RAS 598: Space Robotics and AI (Spring 2025)
March 27, p1~7
ASU ID: 1234000809

5. Key Components

- Input Layer (480x480x3): (Height x Width x Channel)
- Convolutional Layer

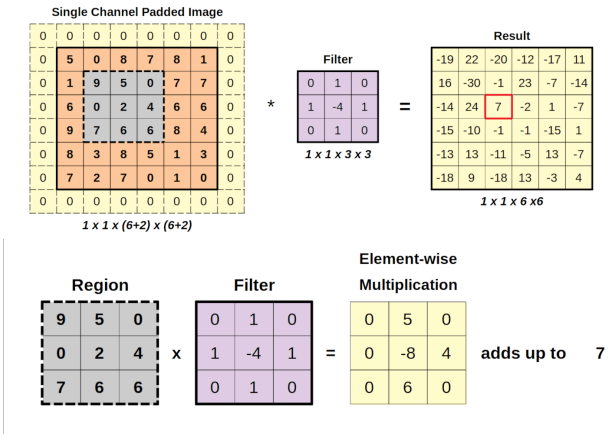
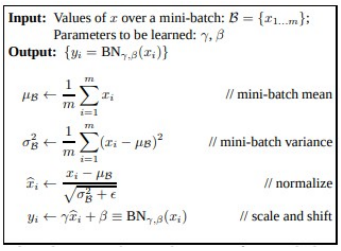


Figure 2 : Convolution Operation

- BatchNormalization



Algorithm: Batch Normalizing Transform, applied to activation x over a mini-batch

- Relu activation function

  Relu performs the non-linear transformation aiming the model to learn more complex tasks.

$$f(x) = max(0, x) \quad (1)$$

- Max pooling

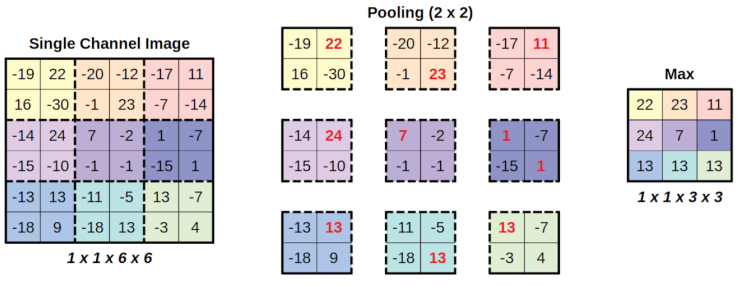


Figure 3. Max Pooling Operation

- Upsampling

  Reconstruction the resolutions of images using Nearest Neighbor (resizing the image)

- Skip Connections
- SoftMax Activation function

  Convert input values into a probability distribution to eligible represent as a probability

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} \quad (2)$$

Z **:** Input values

6. VGG16

VGG16 is a convolution neural network(CNN) based architecture. It is included 16 layers, including 13 convolutional layers and 3 fully connected layers. The model's architecture features a stack of convolutional layers with ReLU(Rectified Linear Unit) followed by MaxPooling layers, with increasing depth.

In this project, pre-trained VGG16 model on the ImageNet dataset is utilized as an encoder component. Additionally, a Decoder component is extended from the last layer of the pre-train VGG16 model.

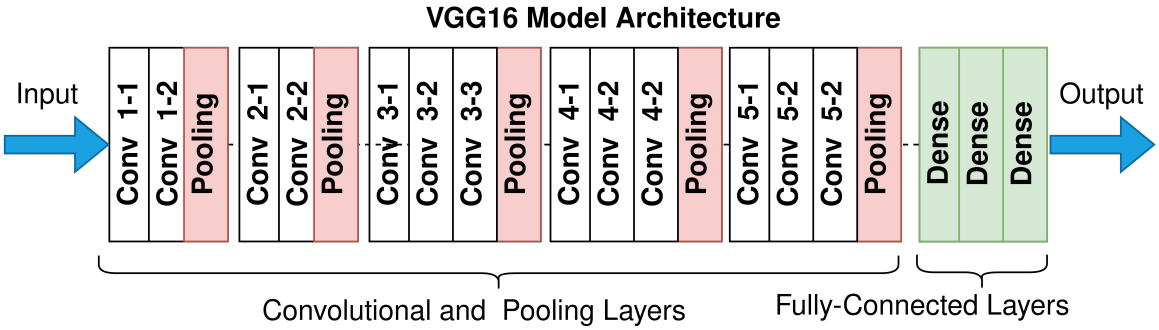


Figure 4. VGG16 Model Architecture

7. Training process

- Cross Entropy Function

  Cross Entropy Function is utilized as a loss function to measures error between the model's predictions and the actual correct data.

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c} \cdot \log(\hat{y}_{i,c}) \quad (3)$$

N : Number of observation

C : Number of category

$y_{i,c}$: Indicator function of the i th observation belonging to the c th category

$\hat{y}_{i,c}$: Probability predicted by the model for i th observation to belong to the c th

- Adam (Adaptive Moment Estimation)

  Adam is utilized to minimize the loss function.

Algorithm 1: *Adam*, our proposed algorithm for stochastic optimization. See section 2 for details, and for a slightly more efficient (but less clear) order of computation. g_t^2 indicates the elementwise square $g_t \odot g_t$. Good default settings for the tested machine learning problems are $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. All operations on vectors are element-wise. With β_1^t and β_2^t we denote β_1 and β_2 to the power t.

Require: α: Stepsize
Require: $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for the moment estimates
Require: $f(\theta)$: Stochastic objective function with parameters θ
Require: θ_0: Initial parameter vector
$m_0 \leftarrow 0$ (Initialize 1st moment vector)
$v_0 \leftarrow 0$ (Initialize 2nd moment vector)
$t \leftarrow 0$ (Initialize timestep)
while θ_t not converged **do**
$t \leftarrow t + 1$
$g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep t)
$m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
$v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment estimate)
$\widehat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)
$\widehat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)
$\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \widehat{m}_t / (\sqrt{\widehat{v}_t} + \epsilon)$ (Update parameters)
end while
return θ_t (Resulting parameters)

Figure 5: Adam optimizer, applied to minimize the loss function

8. Results

In this project, U-net with pre-trained VGG16 model as an encoder is trained with learning rate of 0.0001 and Adam optimizer over categorical cross entropy function.

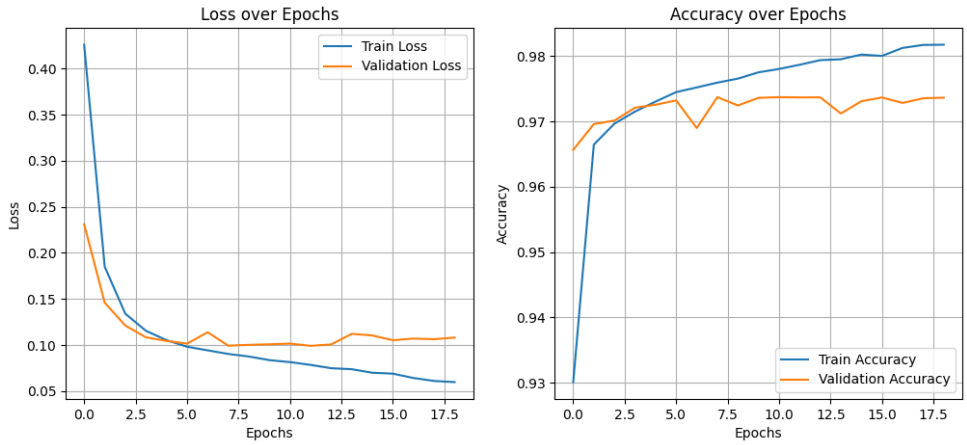
IoU (Intersection over Union) is utilized to evaluate the results.

$$IoU = \frac{correct\ \ mask \cap predicted\ \ mask}{correct\ \ mask \cup predicted\ \ mask} \quad (\ 0 \leqslant IoU \leqslant 1\ ) \ (4)$$

7,966 images were proceeded by using (4) to compute both mean and variance of IoU

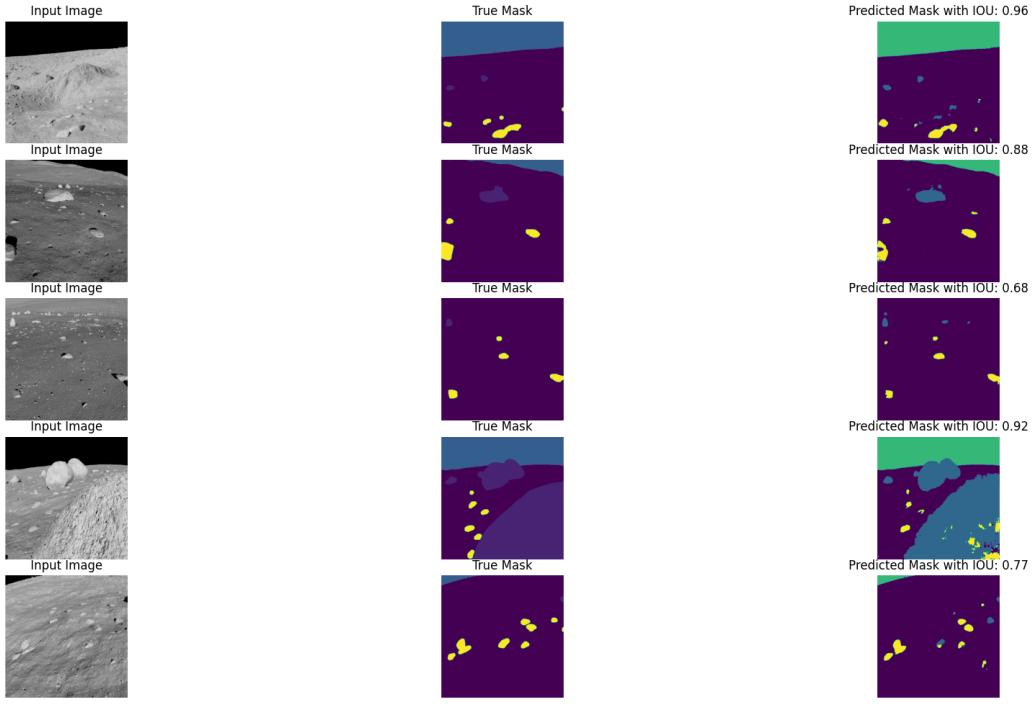
Mean of IoU	**Variance of IoU**
0.8647	

As Figure[6] shows, the loss function is converged to a stable value in only 18 epochs.



(a) (b)

Figure 6. Training Details: (a) Loss over Epochs. (b) Accuracy over Epochs



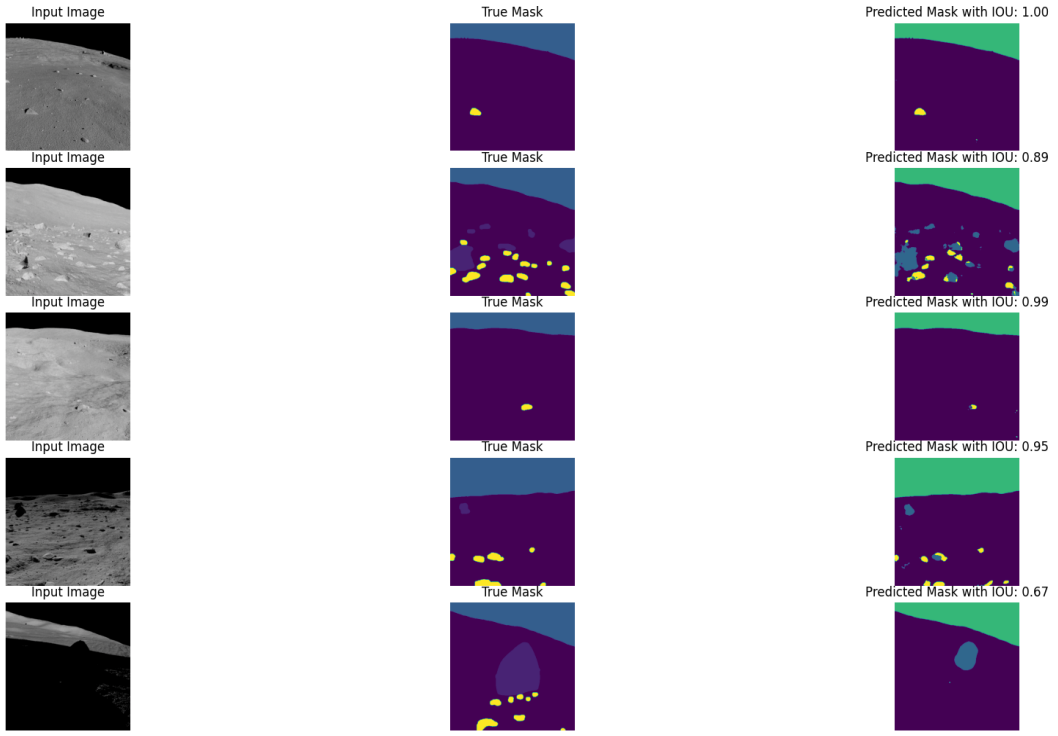


Figure 7. (a) Input Image. (b) Correct Mask. (c) Segmented Image by the model with IoU

9. Conclusions and Final Project

Overall, I believe that semantic segmentation using U-net architecture with background with VGG16 provide good results with mean IoU 0.8647. The model is able to classify between large rock, small rock, sky and ground. However, some of the segmentation results are not up to mark. For example, segmenting the shadow of rocks is considered as sky. Additionally, the model's prediction are stable overall, but there's some variation in performance across different images. Therefore, more fine-tuning is required to be more stable.

Since the results doesn't have depth information, this project can be expanded to semantic segmentation integrated with 3d Gaussian Splatting using terrain_mapping_drone_control(Assignment 3) for the final project.