Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

b-it Bonn-Aachen International Center for Information Technology

Computer Vision Project

Segmentation of Mars Terrain Using Classical Computer Vision Techniques

Ananya Verma

Yogeshkarna Govindaraj

Submitted to Hochschule Bonn-Rhein-Sieg,
Department of Computer Science
in partial fullfilment of the requirements for the degree
of Master of Science in Autonomous Systems

1

Introduction

The rovers sent on the surface of Mars like Curiosity, Perseverance, and Spirit rovers have collected an abundance of data for us to analyze and understand the surface of Mars. With new images coming every day, we can see that the surface of the Red Planet is filled with rocks, sand, and dust.

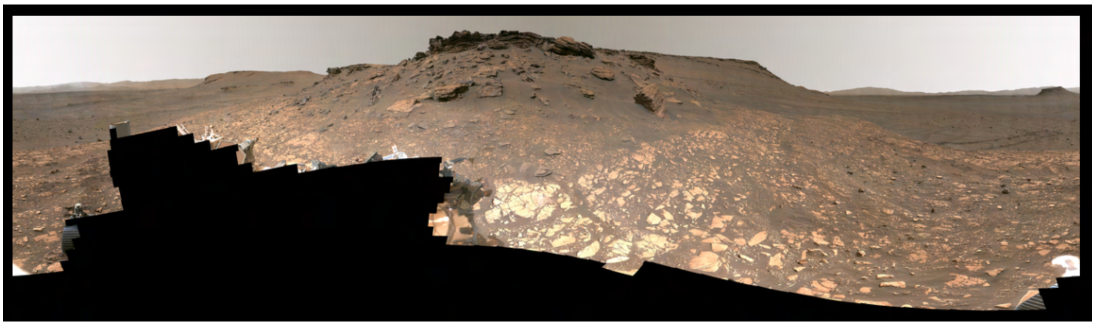


Figure 1.1: Jezero Crater captured by Perseverance rover. Image is reproduced from [1]

The identification of these objects (sand, rocks, bedrock) is necessary not only to analyze the surface of Mars to understand the potential life on the planet but also for the rovers to navigate on the planet [2]. Manually controlling the rover is not ideal given the amount of time the signals take to receive and send to the planet.

One of the major techniques that can be used to segment these images is to use classical methods of Computer Vision like edge detection, RGB segmentation, etc.

Owing to the color and given that the images were taken in different lighting conditions sometimes it is difficult to differentiate the terrain from one another. Hence, segmenting it on the basis of texture and slight variation in color in the presence of shadows would be a challenging task in the field of vision.

For our analysis, we will take the images obtained by the Perseverance rover from the Jezero crater [1].

It is a 2.5-billion-pixel mosaic that combines 1,118 individual frames and is a detailed landscape panorama.

The delta in Mars' Jezero Crater is an area where scientists believe that a river once flowed into a lake and deposited rocks and sediments in a fan shape billions of years ago. Deltas are believed to be the best

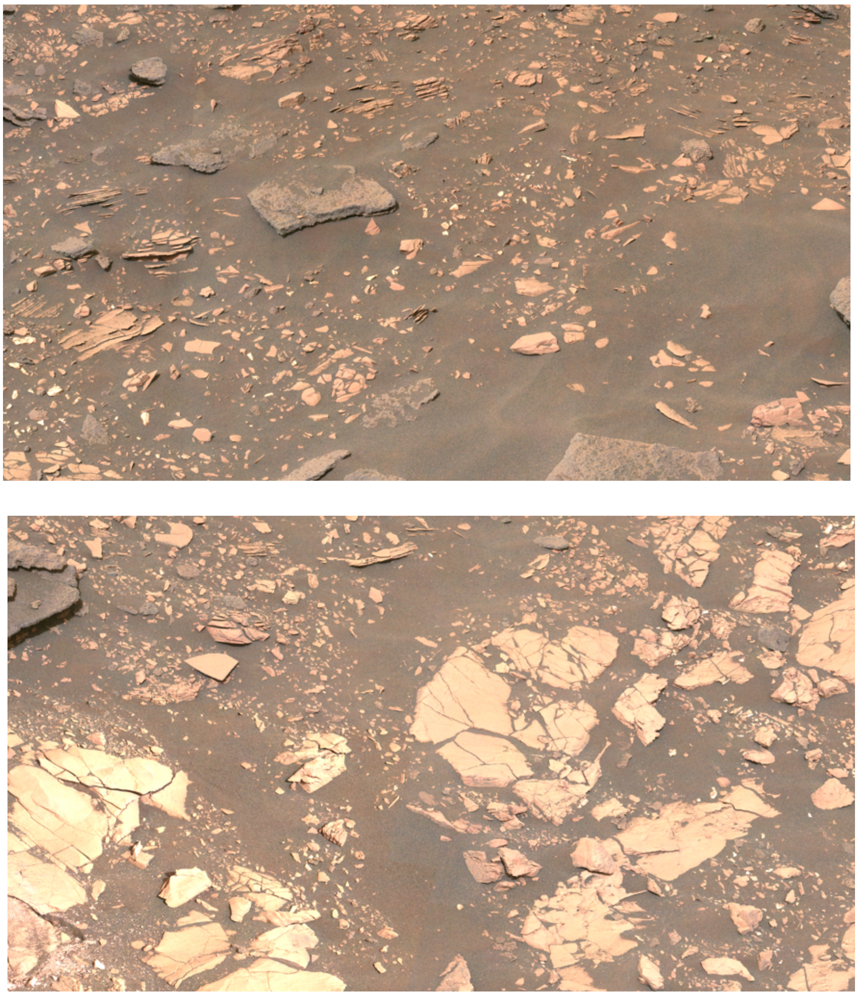


Figure 1.2: Mars terrain image taken from Perseverance rover. Image is reproduced from [1]

places on Mars by scientists to search for potential signs of ancient microbial life. The primary goal of the Perseverance mission since the rover landed in the crater in February 2021 was to study the Jezero delta.

The panorama shows sedimentary rocks. The Perseverance rover has acquired compositional information about this delta.

The goal of this project is to segment the images of Mars terrain captured by the Mars Perseverance rover. The images will be segmented based on the following three classes:

- Rock
- Bedrock
- Sand

An example of the segmentation result can be visualized in Figure 1.3. The big rocks are mostly accompanied by shadows while all the other rocks will be treated as bedrock and the remaining as sand.

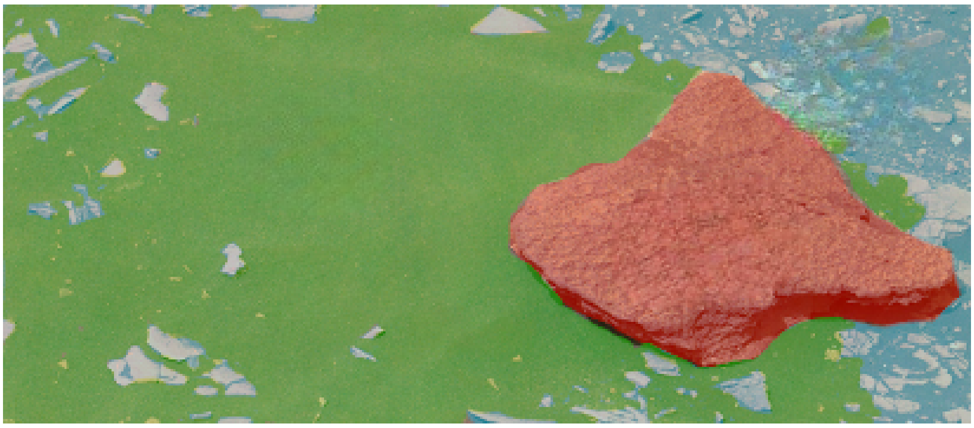


Figure 1.3: Subsection of image in Figure 1.2 with annotation for all three classes (red: rock, blue: bedrock, green: sand). Image is reproduced from [1]

The major challenge would be to identify rocks and bedrock given that there is no uniform color or texture to segment them. In addition, they are also covered by dust and can occlude them or even embed them in the sand [3].

2

Methodology

The panoramic image of Mars terrain captured by the Perseverance rover camera system will be used for the project. The raw data is the 2.5-billion-pixel panoramic image which is created by combining 1,118 individual images of the Jezero crater obtained from [1]. The raw data will be cropped into 4 images of 3840×2160 size images and 2 images have been selected based on high obstacle density.

It is assumed that Mars's surface is only made up of rocks, bedrock, and sand and thus restricted to only these three classes.

2.1 Plan of attack

The plan of attack can be visualized from the flowchart given in Figure 2.4 which is divided into the following three parts:

- **Test data preparation:** The panoramic image [1] will be divided into 2 images of 3840×2160 size and then the subset of these images will be taken to create a test dataset. The test images will be annotated manually.

- **Sand segmentation:** The background elimination will be performed on the subset of images using the following approaches. The sand will be considered as a background here. We have segmented the sand from the bedrock and rock using the method of thresholding, which can be visualized in the Figure 2.1

- **Rock and Bedrock segmentation:** After segmenting the images for class sand, we have removed the pixels belonging to the sand in the image. The following three approaches will be used to segment rock and bedrock:

  – Edge detection in the Figure 2.2

  – K-means method in the Figure 2.3

  This is followed by merging the results obtained from edge detection and the K-Means method to finally segment rocks and bedrocks.

Figure 2.1: Sand segmented using thresholding method

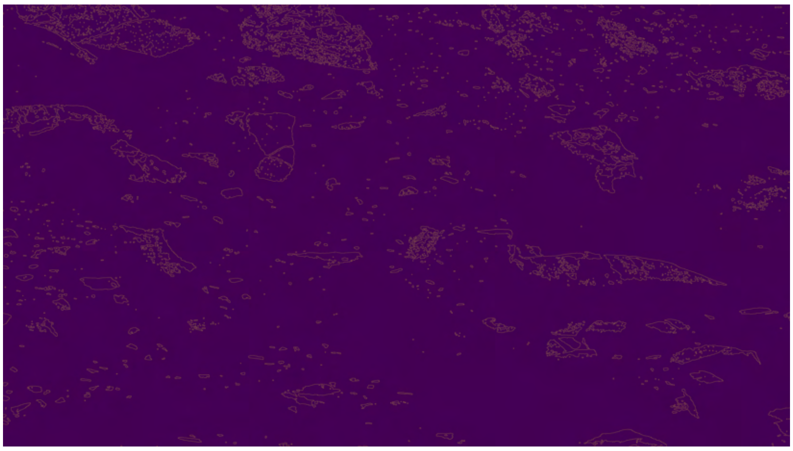


Figure 2.2: Rock and Bedrock segmented using edge detection method

2.2 Thresholding

Thresholding is a process used in image processing and computer vision to separate an image into different classes of pixels, depending on their intensity values. The main idea behind thresholding is to divide an image into two regions: foreground and background, where foreground pixels have an intensity value above a certain threshold, while background pixels have an intensity value below it. The threshold value is usually determined by analyzing the intensity histogram of the image. For our project, we have used Otsu's thresholding which is an adaptive thresholding algorithm that calculates the optimal threshold value that minimizes the intraclass variance of the image.

2.3 Edge Detection

Edge detection is a method used in image processing and computer vision that aims to identify and locate edges within an image. Edges in images can be identified as areas where there is a significant

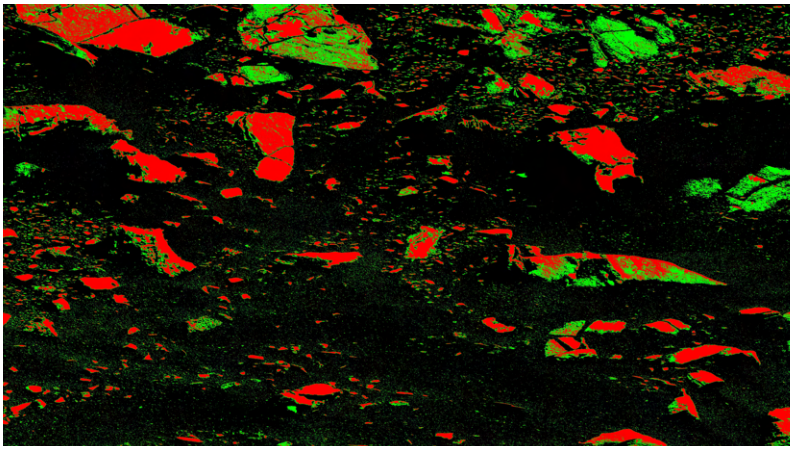


Figure 2.3: Rock and Bedrock segmented using K-means method

change in intensity, color, or texture. The process of edge detection includes analyzing the pixels of an image to locate spots where there is a rapid change in intensity, as these changes correspond to edges.

There are various algorithms available for edge detection, but most of them fall under two main categories: first-order and second-order edge detectors. First-order edge detectors like Sobel and Prewitt operators focus on the difference in intensity between adjacent pixels, while second-order edge detectors such as the Canny operator take into account the overall shape of an edge by analyzing the second derivative of image intensity. For our approach, we have used the canny edge detector, the results of which can be seen in 2.2

2.4 K-Means Algorithm

k-means is an unsupervised machine-learning algorithm commonly used for clustering. The purpose of k-means is to divide a set of n data points into k clusters, in which each data point belongs to the cluster with the closest mean, also known as a centroid, which represents the center of the cluster. The k in the k-means algorithm refers to the number of clusters to be formed and the number of centroids to be generated. For our project, we have initialized k as 3 since three classes had to be segmented.

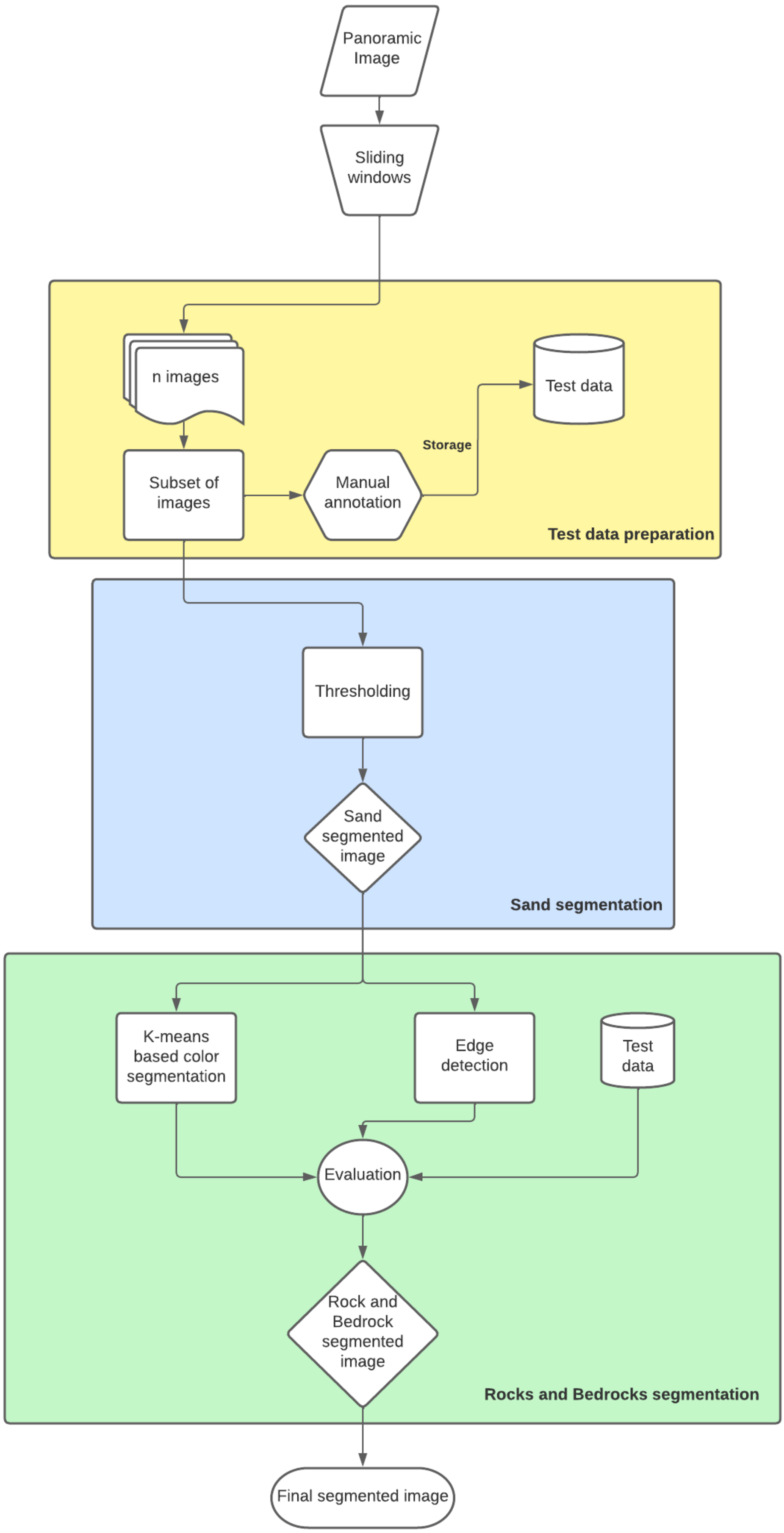


Figure 2.4: Plan of attack

3

Evaluation and Results

In this chapter, we delve into the evaluation methods used to test our approach. Section 3.1 outlines the various metrics used, Section 3.2 provides information on the test dataset, and the results of our tests are outlined in Section 3.3.

3.1 Evaluation Metrics

3.1.1 Accuracy

Accuracy in image processing is a way to evaluate the effectiveness of an algorithm or model in correctly identifying, classifying, or segmenting objects or features in an image. It's commonly used as a performance metric for image processing algorithms.

There are various methods of measuring accuracy in image processing, based on the application at hand. For instance, in image classification, accuracy is usually computed as the ratio of correctly classified images to the total number of images. In image segmentation, accuracy can be measured as the proportion of pixels that are correctly segmented out of the total number of pixels in an image, or as the degree of similarity between the predicted segmentation and the true segmentation.

3.1.2 Recall

Recall is a metric used to predict the performance of object detection or image classification algorithms in computer vision. It calculates the proportion of total positive instances that the algorithm correctly identified. Additionally, it is also known as true positive rate or sensitivity, which is computed as the number of true positive cases (correctly detected) divided by the sum of true positives and false negatives (positive instances that were not identified). It can be represented mathematically as:

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

In simple terms, recall evaluates the proportion of all positive cases that were identified correctly by the algorithm in an image. A recall value of 1 indicates that the algorithm identified all positive cases correctly, while a value of 0 indicates that the algorithm was not able to identify any of the positive instances. In the case of object detection, a high recall value would mean that the algorithm detects objects well, even if it also produces a number of false positives. On the other hand, in image classification, high recall would imply that the model has minimal false negatives, even if it generates some false positives.

3.1.3 Precision

In computer vision, precision is a metric used to evaluate the performance of an object detection or image classification algorithm. It calculates the proportion of total instances that were identified as positive by the algorithm and were actually positive. Also referred as "positive predictive value," it is calculated as the number of true positive cases (correctly identified) divided by the sum of true positives and false positives (incorrectly identified as positive). Mathematically, it can be represented as:

$$Recall = \frac{TruePositives}{TruePositives + FalsePositives}$$

In simpler terms, precision evaluates how many of the instances identified as positive by the algorithm are actually positive. A precision value of 1 implies that all instances identified as positive by the algorithm are truly positive, while a value of 0 implies that none of the instances identified as positive by the algorithm are actually positive. In the context of object detection, a high precision value would mean that the algorithm is good at detecting only the actual objects, even if it misses some. In contrast, in image classification, a high precision value would indicate that the model has minimal false positives, even if it also generates some false negatives.

3.1.4 Confusion Matrix

A confusion matrix, often referred to as an error matrix or a matching matrix, is a tool used to evaluate the performance of a classification algorithm in computer vision and machine learning. It presents a summary of true positives, true negatives, false positives, and false negatives resulting from predictions made by the algorithm, and it is often used along with metrics such as accuracy, precision, recall and F1-score.

Typically, confusion matrix presents four cells which represents the four possible outcomes of a binary classification problem: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

- TP is the number of instances correctly identified as positive.
- TN is the number of instances correctly identified as negative.
- FP is the number of instances incorrectly identified as positive.
- FN is the number of instances incorrectly identified as negative.

Table 3.1: Confusion Matrix for binary classification problem

	Actual Positive	**Actual Negative**
Predicted Positive	TP	FP
Predicted Negative	FN	TN

3.1.5 F1 Score

In computer vision, the F1 score is a widely used metric for evaluating the performance of a classification algorithm. It combines the precision and recall, two key metrics, by taking their harmonic mean, providing a more comprehensive evaluation of the algorithm.

The F1 score is computed by using the following formula:

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

The range of F1-score is between 0 and 1, with a score of 1 indicating perfect precision and recall and a score of 0 indicating the algorithm's failure to identify any positive instances.

The main advantage of F1 score is that it takes into account both precision and recall, and therefore, it can identify the trade-offs between the two. For example, when precision is high, recall is low, and F1-score will be low. Similarly, when recall is high, precision is low and F1-score will be low. In the ideal case, when both precision and recall are high, F1-score will be high as well.

3.2 Test data

As previously discussed, we utilized images obtained by the Perseverance rover from the Jezero crater [1], which were subsequently cropped into smaller images of x×y dimensions. For the purposes of testing, we selected 2 images from this dataset, which can be seen in Figure 3.1.

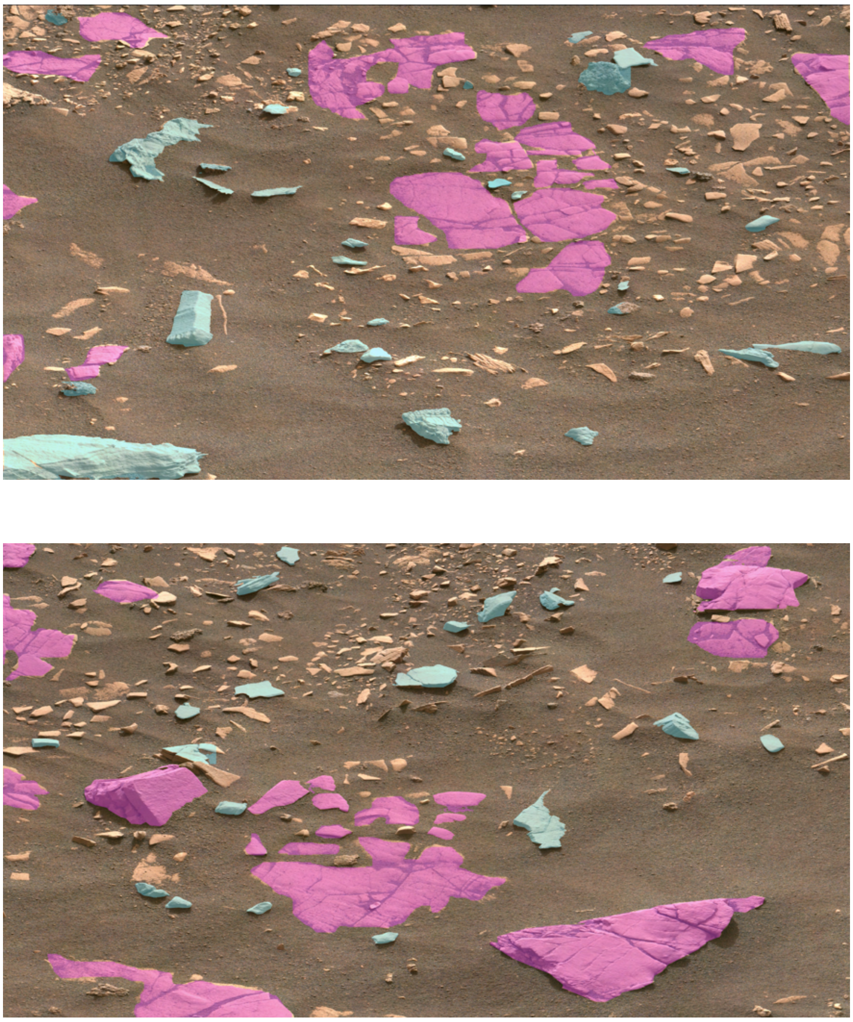


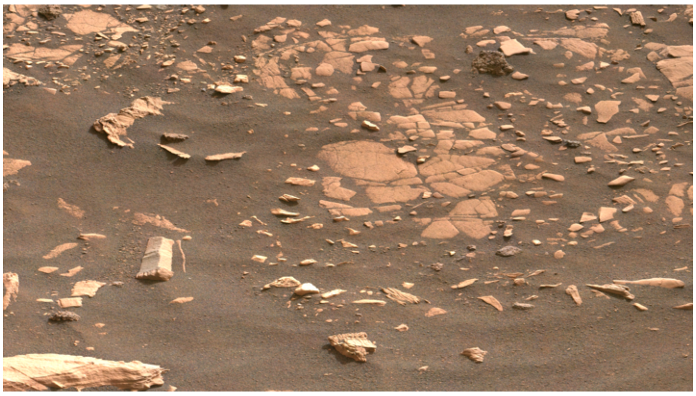
Figure 3.1: Manually annotated test images

We manually labeled the images using the software Labelbox [4]. Our annotations included rocks and bedrock, with the assumption that all other pixels represent sand for the sake of simplicity. Despite successfully annotating the images, we faced some challenges such as difficulty in separating smaller rocks, resulting in inaccuracies in the test dataset. Additionally, exporting the labels was time-consuming and caused difficulties.

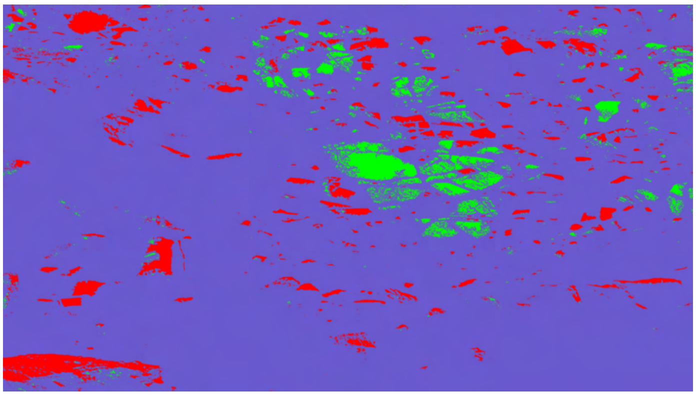
3.2.1 Results

Test image-1

The segmentation results for the three classes in test image 1 can be viewed in Figure 3.2. The scores for various evaluation metrics for the same image can be found in Table 3.2.



(a) Input: test image 1



(b) Output: segmented image 1

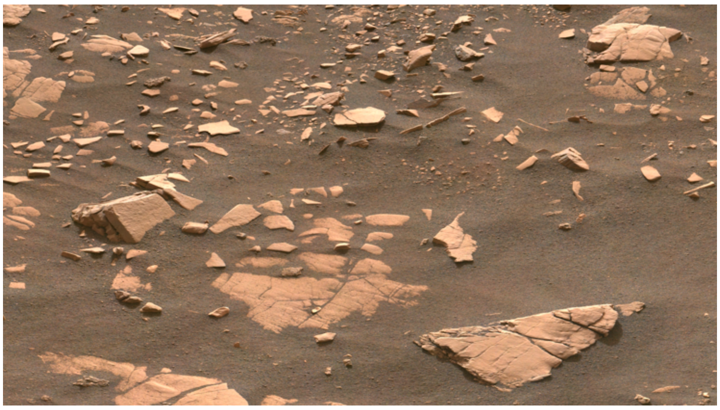
Figure 3.2: Segmentation results for test image 1 for all three classes

Table 3.2: Evaluation scores for test image 1

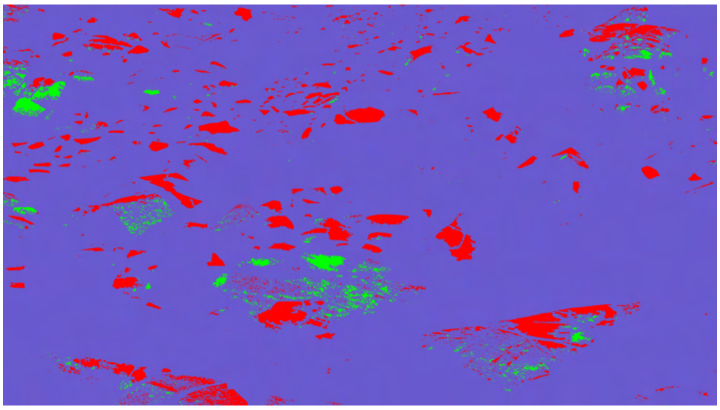
	Test image 1
Accuracy	0.852
Recall	0.852
Precision	0.967
F1 score	0.904
Confusion Matrix	$\begin{bmatrix} 8.69980074e-01 & 4.29001132e-02 & 8.71198125e-02 \\ 1.10376596e-01 & 1.42675644e-04 & 8.89480729e-01 \\ 2.25337036e-01 & 4.35566835e-01 & 3.39096129e-01 \end{bmatrix}$

Test image-2

The segmentation results for the three classes in the test image 2 can be viewed in Figure 3.3. The scores for various evaluation metrics for the same image can be found in Table 3.3.



(a) Input: test image 1



(b) Output: segmented image 1

Figure 3.3: Segmentation results for test image 2 for all three classes

Table 3.3: Evaluation scores for test image 2

	Test image 2
Accuracy	0.844
Recall	0.844
Precision	0.970
F1 score	0.898
Confusion Matrix	$\begin{bmatrix} 8.54204488e-01 & 1.50974969e-02 & 1.30698015e-01 \\ 3.31240946e-02 & 2.17286335e-04 & 9.66658619e-01 \\ 2.14299770e-01 & 2.84677649e-01 & 5.01022581e-01 \end{bmatrix}$

4

Conclusions

4.1 Inference

- From the results, classical computer vision algorithms like thresholding, edge detection, and their combinations have proved to solve the segmentation of terrains with an accuracy greater than 80%.
- The size of images used is of resolution 4K, which has been segmented with an average runtime of 25-30 seconds.
- Sand segmentation through thresholding has been segmented close to zero error apart from the misclassification of shadows of rocks. Also, the bedrock covered with thin layers of sand has been misclassified as sand.
- For segmentation of bedrock from rocks, The difference of light intensity the k-means algorithm performed better.
- Data annotation for segmentation (masking) of 4k images with fine particles like sand is hard and consumes a substantial amount of time. The accuracy of annotation degrades resulting in the wrong annotation.

References

[1] "Detailed Panorama of Mars' Jezero Crater Delta." NASA Mars Exploration. https://mars.nasa.gov/resources/26978/detailed-panorama-of-mars-jezero-crater-delta/ (accessed Nov. 2, 2022).

[2] H. Dunlop, D. R. Thompson, and D. Wettergreen, "Multi-scale features for detection and segmentation of rocks in mars images," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–7.

[3] K. Di, Z. Yue, Z. Liu, and S. Wang, "Automated rock detection and shape analysis from mars rover imagery and 3d point cloud data," *Journal of Earth Science*, vol. 24, 02 2013.

[4] "The data engine for AI ." Labelbox. https://labelbox.com/ (accessed Nov. 10, 2022).