# SPARTIFICIAL INNOVATIONS PRIVATE LIMITED

# ROCKFALL DETECTION ON MOON AND MARS

## ADS-141122-2


**Team Members:**

1. Sakshi Singh
2. Aakash Dubey
3. Aesha Pandya
4. Rithesh

**Name of guides:**

1. Tanisha Medewala
2. Sudhir Sajan

# INDEX

# 1. Abstract

This study aims in detecting rockfall on Mars and Moon. This examination will help in understanding the development of the surfaces of Mars and the Moon, as well as studying their aged and ongoing endogenic and explicate conditioning. For mapping, high-resolution space-borne picture data of Mars and the Moon are employed. Because mortal mapping of displaced boulders is hamstrung and slow, performing in a tiny total number of counter plotted features, a deep learning-driven technique is used to detect and map rockfall trails in Hi-RISE (High-resolution imaging scientific experiment) imagery.

# 2. Introduction

Mars has recently gained transnational attention. Many examinations are carried out on the surface in order to comprehend its nature. One method for understanding the landscape is to identify rockfalls. Like polar ice or dust avalanches, rock fall is a mass-wasting process. The moon is also a good place to watch for falling rocks. Understanding and observing rockfalls provides us with precious information similar to surface strength, surface characteristics, and fall rate. This allows us to forecast the future much more accurately and easily.

As several rockfalls occur over time, mapping those using only satellite pictures is extremely grueling. There are still numerous areas where satellites are unable to reach and cannot transmit data. Not only do falling boulders supply valuable information, but seeing craters has similar issues. Hi-RISE photos are very useful in this endeavor since they give high-resolution images that may be processed further. Human analysts evaluated all rockfall images and attempted to map all rockfall sites using the traditional approach. However, this method is very inefficient and sluggish. Complex subjects are difficult to notice.

# 3. Related Research

The problem statement that we have is how far the rockfall sites can be located efficiently and precisely using CNN. The Hi-RISE images of Mars and the Moon are the only source of data we have. Hi-RISE images are used because they have more pixels than standard images. These images may or may not show rock falls. Some may have one, while others may have several. The crucial task is to create a model to analyze the images and determine whether or not they include rock fall locations. An appropriate model is to be used to do the interpretation task precisely. The terrains of Mars and the Moon are distinct. As a result, the model must become acquainted with nearly all of the implicit characteristics. In order to get effective outcomes, all images intended for training must be appropriately fed.

### 3.1 Dataset Description

Hi-RISE images are the best-resolution image. Here we discuss Mars and Moon rockfall. Mars has a huge area which is not covered by the camera so HIRISE could capture images only in a limited area. Data for Mars and Moon has been fetched from an online source called “Rockfall Detection on Mars” and “Rockfall Detection on Moon”. The data set consists Mars and Moon directory with test _images, test labels, train_ images, and train labels. There are more than 1000 positive and negative images to train the model and 90 positive images and 20 negative images to test the label. There is a .csv file for each label folder included with the name of the images, and co-ordinates of detected rockfall.

```python
#Reading Data
mars_train =pd.read_csv("drive/My Drive/mars/train_labels/train_labels_ma.csv")
mars_train.head()
mars_train
```

| | IMAGE_ID | X1 | Y1 | X2 | Y2 | label |
|---|---|---|---|---|---|---|
| 0 | val_1.jp2 | 370.0 | 220.0 | 394.0 | 243.0 | rockfall |
| 1 | val_1.jp2 | 311.0 | 391.0 | 342.0 | 423.0 | rockfall |
| 2 | val_10.jp2 | 117.0 | 151.0 | 144.0 | 182.0 | rockfall |
| 3 | val_11.jp2 | 236.0 | 112.0 | 270.0 | 178.0 | rockfall |
| 4 | val_11.jp2 | 259.0 | 439.0 | 279.0 | 477.0 | rockfall |
| ... | ... | ... | ... | ... | ... | ... |
| 1295 | neg43.JP2 | NaN | NaN | NaN | NaN | NaN |
| 1296 | neg44.JP2 | NaN | NaN | NaN | NaN | NaN |
| 1297 | neg45.JP2 | NaN | NaN | NaN | NaN | NaN |
| 1298 | neg46.JP2 | NaN | NaN | NaN | NaN | NaN |
| 1299 | neg47.JP2 | NaN | NaN | NaN | NaN | NaN |

1300 rows × 6 columns

Fig 1. Mars Data set (Mars train data set with IMAGE_ID label as Rockfall)

```python
#Reading Data
moon_train = pd.read_csv('drive/My Drive/mars/train_labels/train_labels_ma.csv')
moon_train
```

| | IMAGE_ID | X1 | Y1 | X2 | Y2 | label |
|---|---|---|---|---|---|---|
| 0 | val_1.jp2 | 370.0 | 220.0 | 394.0 | 243.0 | rockfall |
| 1 | val_1.jp2 | 311.0 | 391.0 | 342.0 | 423.0 | rockfall |
| 2 | val_10.jp2 | 117.0 | 151.0 | 144.0 | 182.0 | rockfall |
| 3 | val_11.jp2 | 236.0 | 112.0 | 270.0 | 178.0 | rockfall |
| 4 | val_11.jp2 | 259.0 | 439.0 | 279.0 | 477.0 | rockfall |
| ... | ... | ... | ... | ... | ... | ... |
| 1295 | neg43.JP2 | NaN | NaN | NaN | NaN | NaN |
| 1296 | neg44.JP2 | NaN | NaN | NaN | NaN | NaN |
| 1297 | neg45.JP2 | NaN | NaN | NaN | NaN | NaN |
| 1298 | neg46.JP2 | NaN | NaN | NaN | NaN | NaN |
| 1299 | neg47.JP2 | NaN | NaN | NaN | NaN | NaN |

Fig 2. Moon Data set (Moon train dataset with IMAGE_ID label as Rockfall)

# 4. Proposed Solution

The primary goal is to classify the labels as non-rockfall and rockfall. Data set should be divided into training data and test data. Then once the region is detected or images are cropped out from the original image file, the cropped images turned out to be different shapes and sizes, so image resizing process is required to have same image size file. Once the data set ID finalized with all the count of data, denoising process for smoothen the images. Gaussian blur effect which help to remove any noise from the images. Next augmentation, which actually find the region of rock-fall in image, then it create bounding box around and try to crop the images surrounded by bounding box and this gives us a positive rock-fall images with small cut out.

For image processing from data set for transformation, Numpy was used. In data transformation, label.csv file that adding a row of heading of each column such as “Image Id”,”x1”,”y1”,”x2”,”y2”, “label name”, this change is helpful to fetch the data from csv file by using its column name. The .csv file as contain data with label name as “rockfall”.to classify the other image, it was named as “non-rock fall” through python coding. We create a data from rock fall images that how much duplicate images we have in our data set, and there is 778 duplicate images out of 1000 positive images. As we can see in Fig 3.

```
#List of duplicate Image IDs i.e. 778 rows
duplicate_mars = mars_train1[mars_train1.duplicated(['IMAGE_ID'], keep = False)]
duplicate_mars
```

| | IMAGE_ID | X1 | Y1 | X2 | Y2 | label |
|---|---|---|---|---|---|---|
| 0 | val_1.jp2 | 370.0 | 220.0 | 394.0 | 243.0 | rockfall |
| 1 | val_1.jp2 | 311.0 | 391.0 | 342.0 | 423.0 | rockfall |
| 3 | val_11.jp2 | 236.0 | 112.0 | 270.0 | 178.0 | rockfall |
| 4 | val_11.jp2 | 259.0 | 439.0 | 279.0 | 477.0 | rockfall |
| 5 | val_12.jp2 | 169.0 | 350.0 | 191.0 | 387.0 | rockfall |
| ... | ... | ... | ... | ... | ... | ... |
| 995 | I94.JP2 | 169.0 | 73.0 | 250.0 | 210.0 | rockfall |
| 996 | I94.JP2 | 393.0 | 187.0 | 455.0 | 290.0 | rockfall |
| 997 | I94.JP2 | 369.0 | 140.0 | 408.0 | 203.0 | rockfall |
| 998 | I95.JP2 | 468.0 | 303.0 | 517.0 | 372.0 | rockfall |
| 999 | I95.JP2 | 708.0 | 484.0 | 801.0 | 664.0 | rockfall |

778 rows × 6 columns

Fig 3. Total Duplicate images of Mars Data

Construct X_label and Y_label. The image files are stored in X_label, and the ultimate contains the equals of the rockfall spots in the images as well as whether or not the images contain rockfall spots. Y_label remains unchanged. Still, the number of accessible images in the X_label is mainly lower than the number of rows in the Y_label. This causes a data imbalance. This is due to the possibility of one or further rockfall locales being included in a single image. As a result, the equality of such locales must be specified individually. However, image_1 has two rockfall locales; the Y_label must have two rows of the same image with two distinct coordinates. In reality, the Y_label had 1000 rows but only 778 images in the X_label. As it turns out, this is one of the hurdles that must be overcome. The images that aren't duplicates are placed into the X_label. The duplicate images are reproduced in the same fashion, but the number of times is determined by the number of rockfall locations present. The X_label now contains all of the non-duplicate images as well as the duplicate images that match the Y_label..

```python
#Creating the csv file, Dropping the missing values from the data and Reading the data
mars_train1 = mars_train.dropna()
mars_train1.to_csv('ylabel_mars.csv')

!cp ylabel_mars.csv "/content/drive/MyDrive/mars/y_label_mars"

y_label_temp = pd.read_csv("/content/drive/MyDrive/mars/y_label_mars/ylabel_mars.csv")

y_label_temp = mars_train1.loc[:, ~mars_train1.columns.str.contains('^Unnamed')]

y_label_temp
```

| | IMAGE_ID | X1 | Y1 | X2 | Y2 | label |
|---|---|---|---|---|---|---|
| 0 | val_1.jp2 | 370.0 | 220.0 | 394.0 | 243.0 | rockfall |
| 1 | val_1.jp2 | 311.0 | 391.0 | 342.0 | 423.0 | rockfall |
| 2 | val_10.jp2 | 117.0 | 151.0 | 144.0 | 182.0 | rockfall |
| 3 | val_11.jp2 | 236.0 | 112.0 | 270.0 | 178.0 | rockfall |
| 4 | val_11.jp2 | 259.0 | 439.0 | 279.0 | 477.0 | rockfall |
| ... | ... | ... | ... | ... | ... | ... |
| 995 | I94.JP2 | 169.0 | 73.0 | 250.0 | 210.0 | rockfall |
| 996 | I94.JP2 | 393.0 | 187.0 | 455.0 | 290.0 | rockfall |
| 997 | I94.JP2 | 369.0 | 140.0 | 408.0 | 203.0 | rockfall |
| 998 | I95.JP2 | 468.0 | 303.0 | 517.0 | 372.0 | rockfall |
| 999 | I95.JP2 | 708.0 | 484.0 | 801.0 | 664.0 | rockfall |

1000 rows × 6 columns

Fig 4. Drop the missing values

Followed by the pre-processing of the given images. The first stage in the pre-processing procedure is to resize the images to a consistent dimension. Because each image may have various sizes, resizing is needed. The construction of the bounding box comes next. Following the development of the bounding box, each image coordinates must be normalized. Co-ordinate normalization improves consistency and supports effective learning.

For the coding part, Google Colab and Kaggle used. The purpose of using it was a) providing more computational power than local machine, b) Data and programme syncing feature and c) no need for setup preparation. By using the above platforms, data was imported into the Google Drive. Data was downloaded from the authentic source to the local machine and then the .zip file was unzipped in local then uploaded in Google drive.

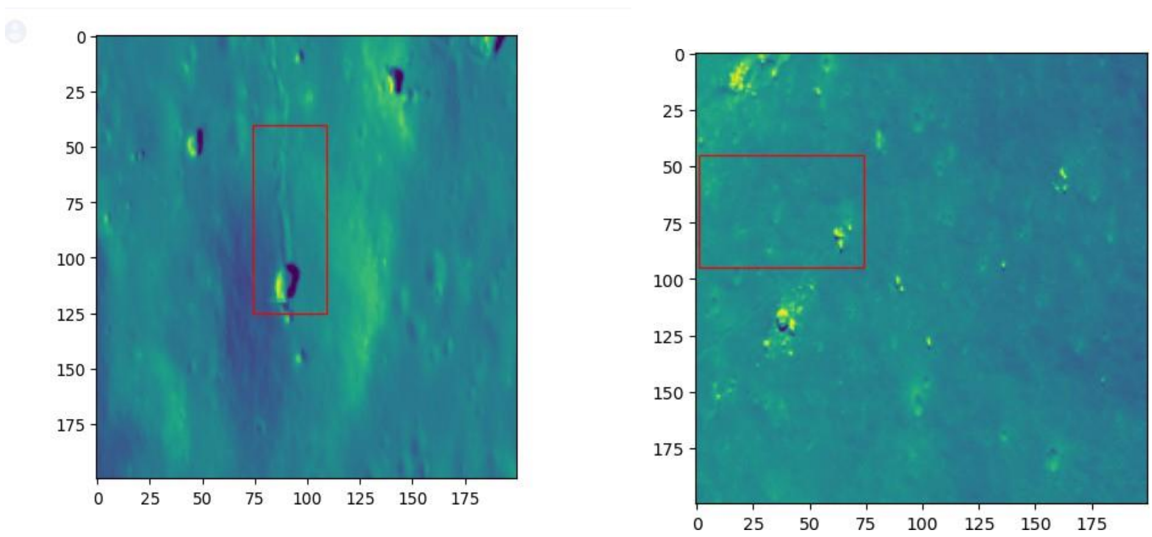



Fig 5. After Image Processing (Moon Rockfall)

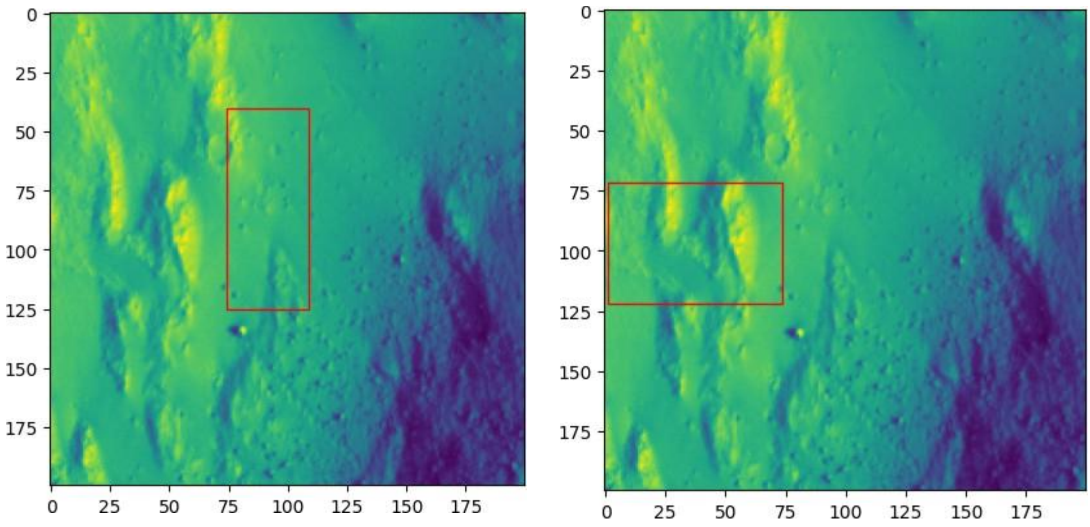



Fig 6. After Image Processing (Mars Rockfall)

The pre-processing phase is concluded, and the modelling phase has began. CNN is employed as an image processing model in this work to identify rockfalls by utilizing Hi-RISE images. GAN is the model utilized in this research. To locate the rockfall in that image is examined pixel by pixel. Since there can be more than one rockfall in a single image, each rockfall in that image is stored in a rectangular box. After that, it's just a matter of classifying whether or not it contains a rockfall site, and if so, where and how many. Classification takes place in two stages: training and testing. Training the model occurs when the model learns to find places where there is a rockfall. Testing the model: the model is tested by feeding images of a rockfall and then confirming whether the model can detect the rockfall and how precisely it can detect it.

Tensor flow was employed, and the sequential model was enforced. Because VGG16 has a lower accuracy than the Sequential model, the Sequential model is used. The model has now been trained. The model is tested using test images, and the accuracy obtained is 94.00% for the moon and 80.13% for Mars.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_4 (Conv2D)           (None, 198, 198, 1024)    10240

 max_pooling2d_4 (MaxPooling  (None, 99, 99, 1024)     0
 2D)

 conv2d_5 (Conv2D)           (None, 97, 97, 512)       4719104

 max_pooling2d_5 (MaxPooling  (None, 48, 48, 512)      0
 2D)

 conv2d_6 (Conv2D)           (None, 46, 46, 256)       1179904

 max_pooling2d_6 (MaxPooling  (None, 23, 23, 256)      0
 2D)

 conv2d_7 (Conv2D)           (None, 21, 21, 128)       295040

 max_pooling2d_7 (MaxPooling  (None, 10, 10, 128)      0
 2D)

 flatten_1 (Flatten)         (None, 12800)             0

 dense_3 (Dense)             (None, 128)               1638528

 dense_4 (Dense)             (None, 64)                8256

 dense_5 (Dense)             (None, 4)                 260

=================================================================
Total params: 7,851,332
Trainable params: 7,851,332
Non-trainable params: 0
```

Fig 7. CNN Model

## 5. Experimental Results & Discussion

The number of epochs for Sequential model is chosen for this model was 4 with batch size of 4. Below from Fig 8. see that the moon data set has 94% accuracy after training the data. In Fig 9. for mars data set has been 80.13% accuracy after training the data.

```
history = model.fit(X_train, y_train, epochs=4, batch_size = 4)

Epoch 1/4
188/188 [==============================] - 2323s 12s/step - loss: 109568.1406 - accuracy: 0.9320
Epoch 2/4
188/188 [==============================] - 2315s 12s/step - loss: 109567.6016 - accuracy: 0.9400
Epoch 3/4
188/188 [==============================] - 2298s 12s/step - loss: 109567.5859 - accuracy: 0.9400
Epoch 4/4
188/188 [==============================] - 2290s 12s/step - loss: 109567.5547 - accuracy: 0.9400
```

Fig 8. Accuracy after training (Moon)

```
history = model.fit(X_train, y_train, epochs=4, batch_size = 4)

Epoch 1/4
188/188 [==============================] - 4662s 25s/step - loss: 14848.2109 - accuracy: 0.8013
Epoch 2/4
188/188 [==============================] - 4519s 24s/step - loss: 14848.1035 - accuracy: 0.8013
Epoch 3/4
188/188 [==============================] - 4554s 24s/step - loss: 14848.1133 - accuracy: 0.8013
Epoch 4/4
188/188 [==============================] - 4540s 24s/step - loss: 14848.1094 - accuracy: 0.8013
```

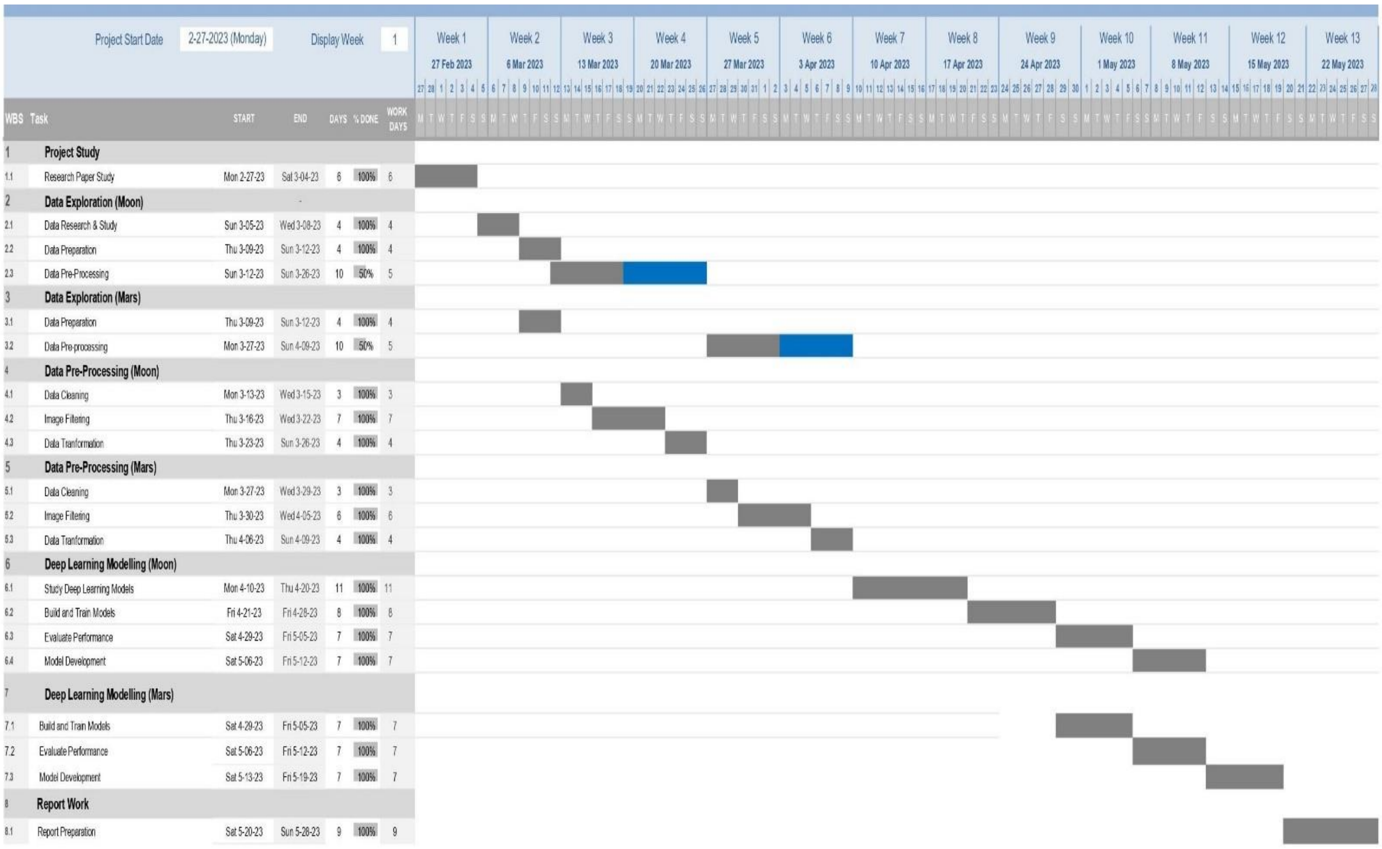
Fig 9. Accuracy after training (Mars)

## 6. Conclusion & Future Work

We find that employing the Guassian blur effect to identify the labels rockfall and non-rockfall eliminated any noise from the photos. Image resizing is crucial in Pre-processing, and CNN is utilized in the modeling step to recognize rockfall. Then the GAN model was utilized to locate the rockfall picture pixel by pixel. For testing and training, an image sequential model yielded 94% accuracy for the moon and 80.13% accuracy for Mars, which is greater than VGG16.In this project train time data augmentation has been performed, but not in test. Another important action is training the model with great number of positive images in dataset and negative images.

As we see that so many ways to enhance and apply many further other models. By checking many further layers of CNN which can be favorable for project. Adding further layers can have two profits. More pooling layers with same amount of information will be applicable further to the ground truth, also start layers trained fluently by lines whereas deeper layer can learn more complex features. More filters will help to achieve further accuracy into data. By using of different technique for pre-processing the images. Many other ways can be applied for the synthetic data preparation

# 7. Gantt Chart



# 8. References


a. https://www.kaggle.com/datasets/yash92328/rockfall-detection-on-mars.

b. https://www.kaggle.com/datasets/yash92328/rockfall-detection-on-moon.

c. Dutta Bhowmik, Soumi. *"Rockfall Detection on Mars using Deep Learning Algorithm"* PhD diss., Dublin, National College of Ireland, 2022. https://norma.ncirl.ie/id/eprint/6116.

d. Bickel, Valentin Tertius, Lukas Mandrake, and Gary Doran. *"A labelled image dataset for deep learning-driven rockfall detection on the Moon and Mars."* Frontiers in Remote Sensing 2 (2021): 640034.

e. V. T. Bickel, S. J. Conway, P. -A. Tesson, A. Manconi, S. Loew and U. Mall, *"Deep Learning-Driven Detection and Mapping of Rockfalls on Mars,"* in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 13, pp. 2831-2841, 2020, doi: 10.1109/JSTARS.2020.2991588.