README.md

Mars-Curiosity-Rover-Image-Classification-Project

Image Classification on Mars with 24 different classes

Overview

This is Image Classification project about Mars curiosity rover. Contains 6691 images collected 2012 to 2015. In this project, NASA or any other space companies who work on mars can find image classification model about mars. This could help the business to deploy our machine learning model on the future rovers to make better and better decisions by itself. NASA has two future missions shows on website; first Mars Sample Return, second ExoMars 2022 Rover and Surface Platform. Our model could deploy on that missions.

Business Problem

Our aim is in this project is to image classification about Mars. Images collected by Curiosity Rover which launched Nov. 26, 2011 and landed on Mars at Aug. 5, 2012. Curiosity set out to answer the question: Did Mars ever have the right environmental conditions to support small life forms called microbes? We are going to classify images collected by curiosity rover to help future projects. Our aim to make machine learning model and deploy at the future rovers to make future rover more effective decisions with artificial intelligence depend on mission.

Dataset

The dataset provided on https://www.kaggle.com/datasets/brsdincer/mars-surface-and-curiosity-image-set-nasa .

Dataset set consists of 6691 images about Mars and rover's itself. Images collected between August 2012 to July 2015. Images are 24 different categories includes horizon , surface and rover's instruments(like cameras, sensors, chemistry tools etc.). We can see some them at below.

Horizon

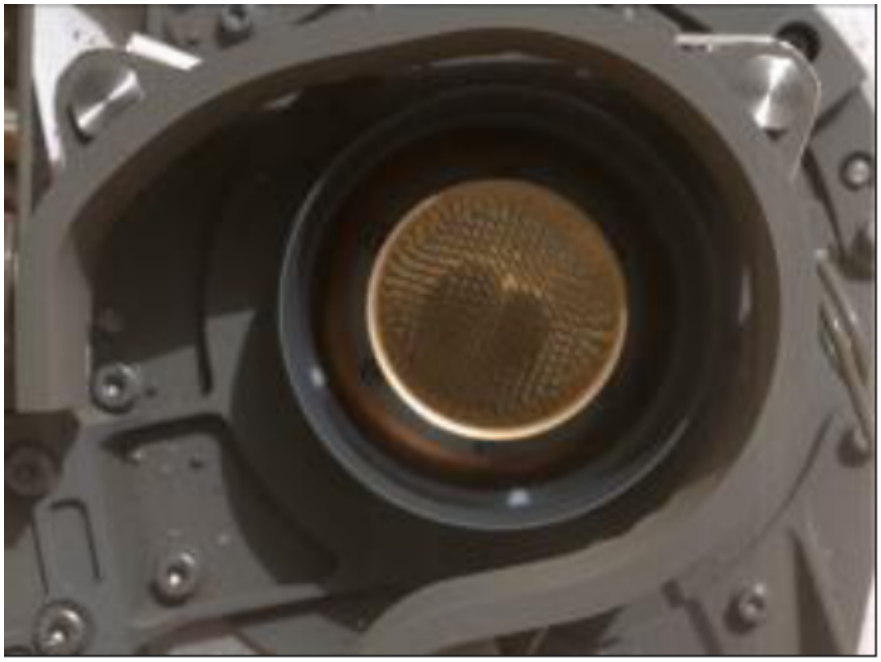
Surface

MASTCAM

The Mast Camera, or Mastcam for short, takes color images and color video footage of the Martian terrain. The images can be stitched together to create panoramas of the landscape around the rover.

CHEMIN

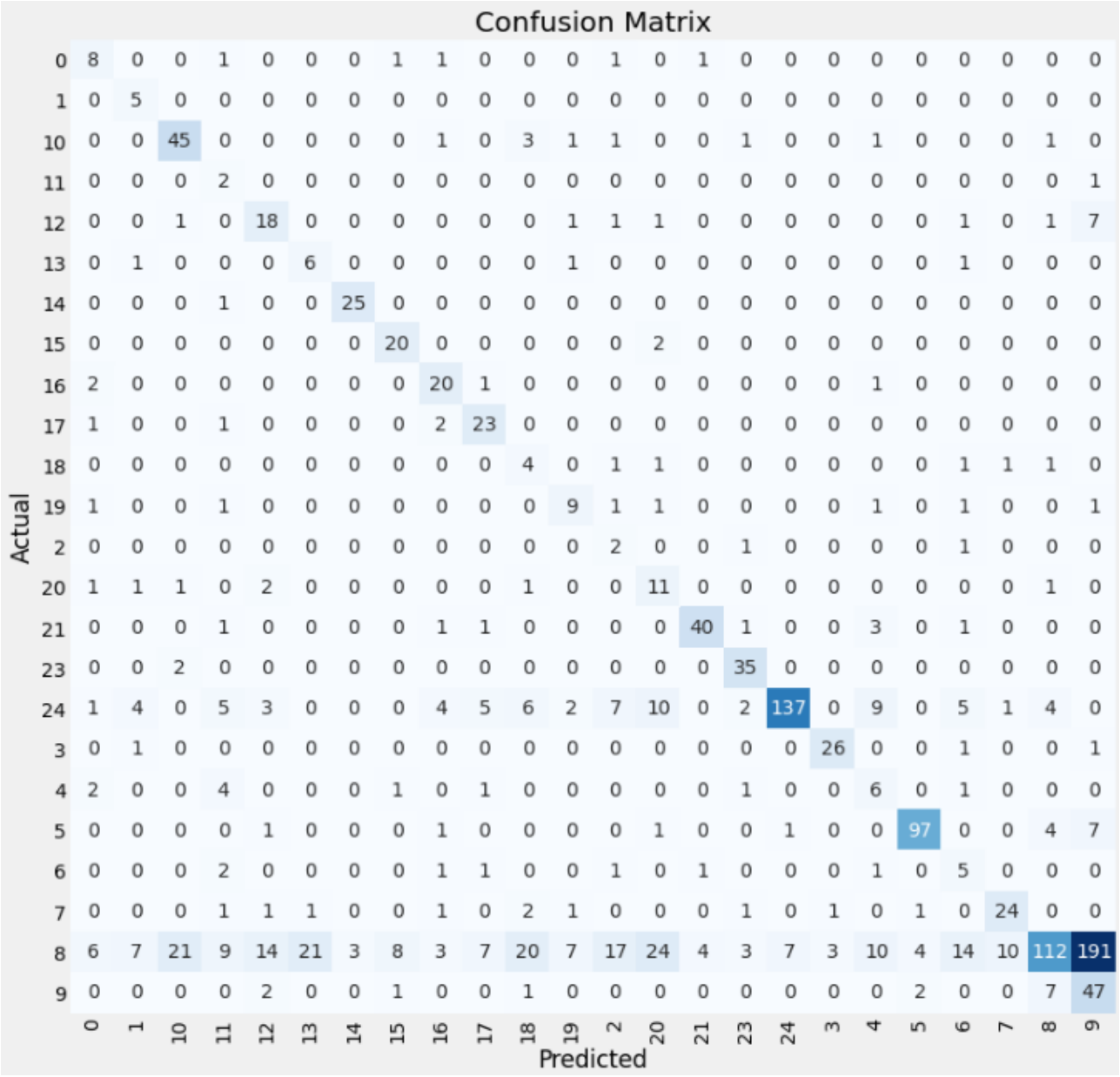
The Chemistry and Mineralogy instrument, or CheMin for short, performs chemical analysis of powdered rock samples to identify the types and amounts of different minerals that are present.



Modeling

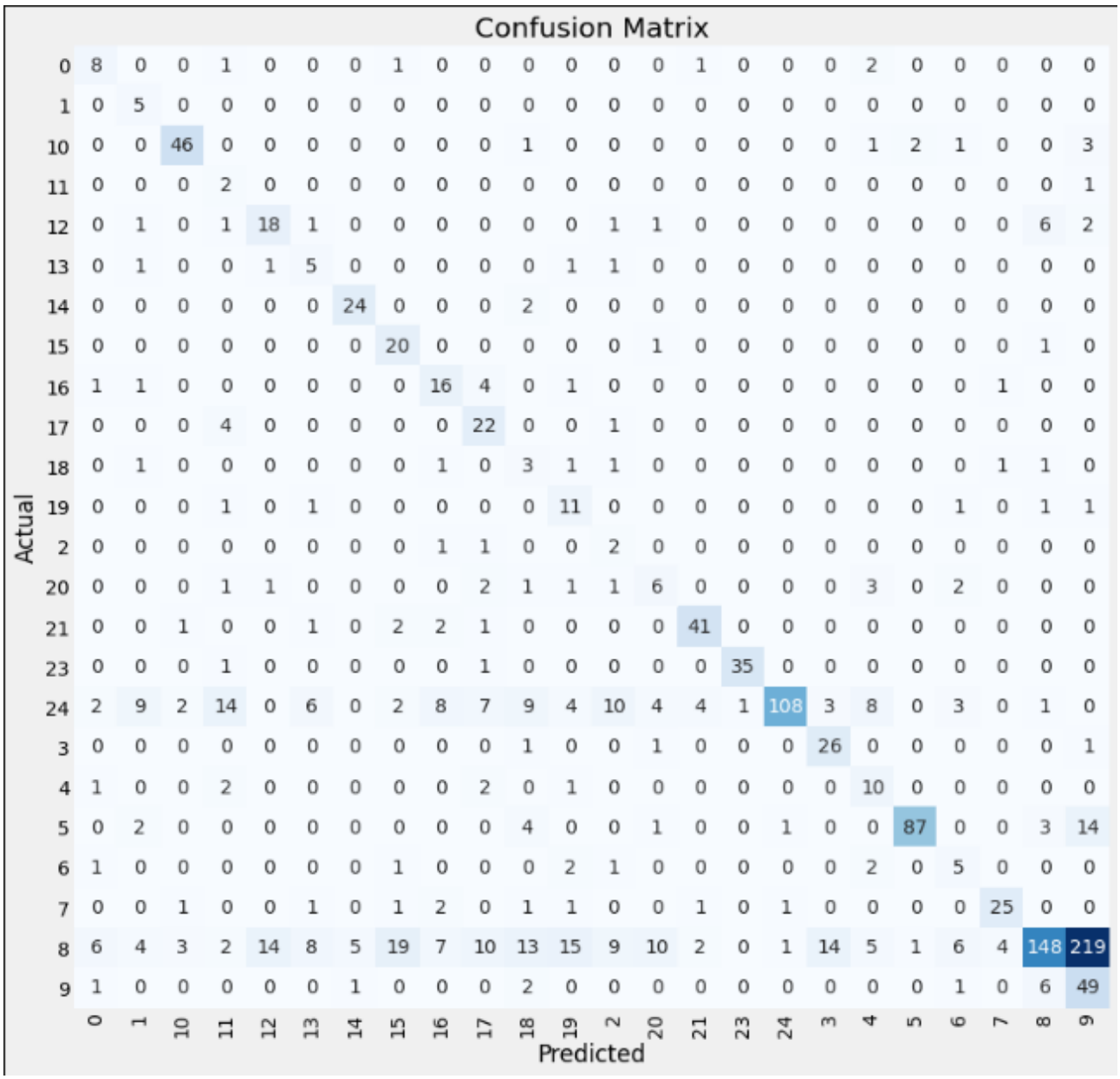
The main metric that I would be using to assess my models performance is `f-1 score` . F-1 score explain how good the quality of predictions are and how completely we've predicted labels from dataset. We wouldn't look at accuracy score because it would be misleading for our specific project. Because accuracy generally good for balanced classes and if every classes importances the same. We are goin to look at F-1 Score because it is harmonic mean of precision and recall scores what exactly need for this project. Which is for this project every class important for us.

First Model



First model f-1 score is %54.29 .

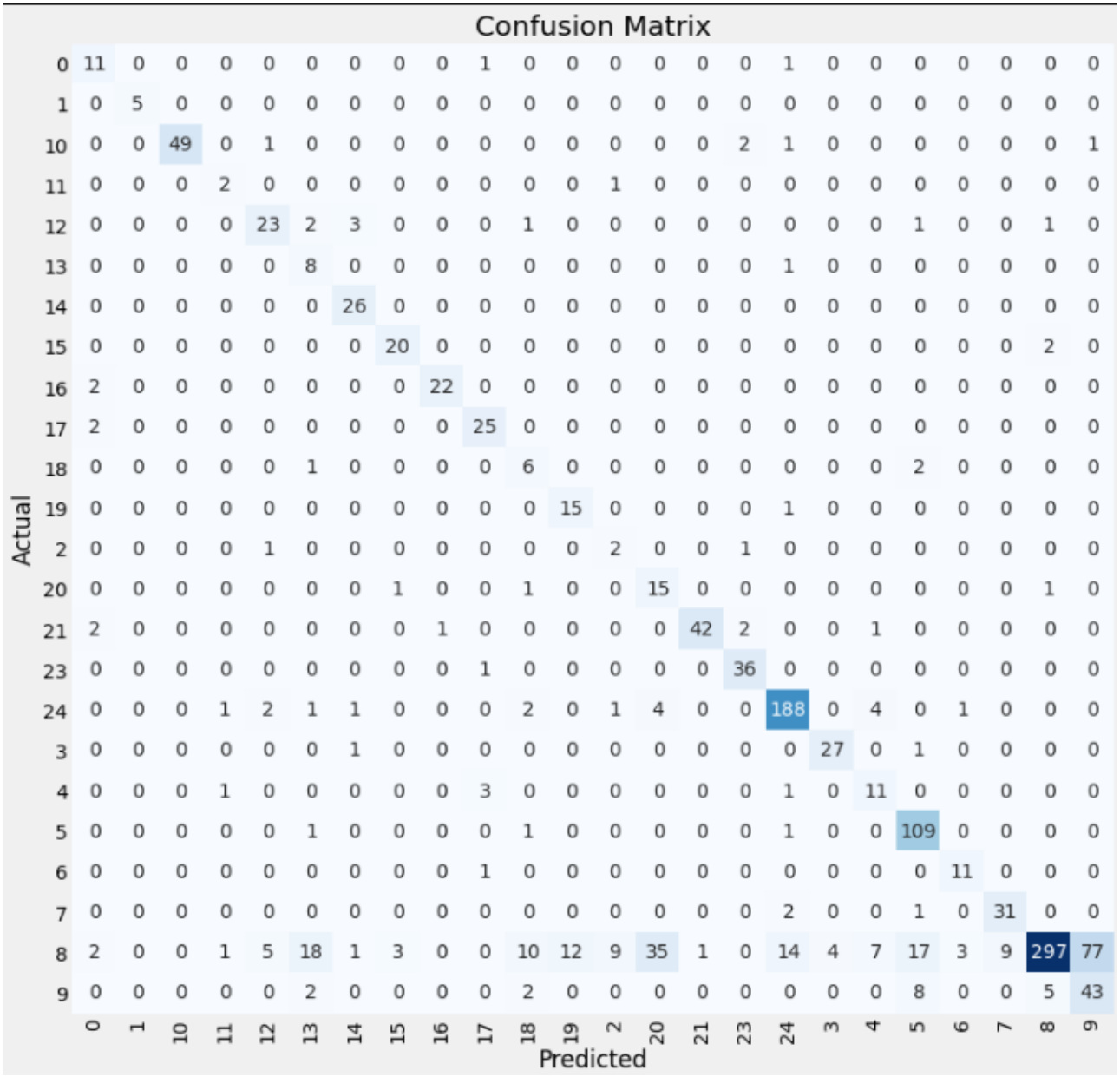
Second Model

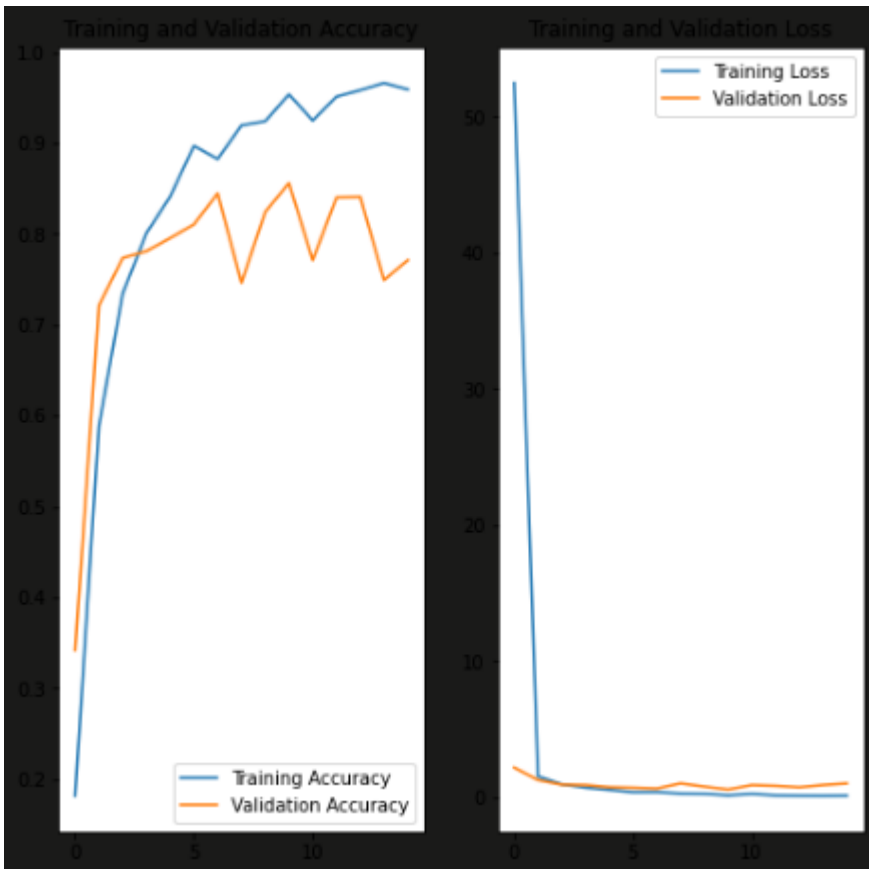


Second model f-1 score is %53.92 .

Third Model

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_7 (Conv2D)           (None, 256, 256, 32)      896

 max_pooling2d_5 (MaxPooling  (None, 128, 128, 32)     0
 2D)

 conv2d_8 (Conv2D)           (None, 128, 128, 32)      9248

 max_pooling2d_6 (MaxPooling  (None, 64, 64, 32)       0
 2D)

 conv2d_9 (Conv2D)           (None, 64, 64, 64)        18496

 max_pooling2d_7 (MaxPooling  (None, 32, 32, 64)       0
 2D)

 dropout_3 (Dropout)         (None, 32, 32, 64)        0

 flatten_2 (Flatten)         (None, 65536)             0

 dense_4 (Dense)             (None, 128)               8388736

 dense_5 (Dense)             (None, 24)                3096

=================================================================
Total params: 8,420,472
Trainable params: 8,420,472
Non-trainable params: 0
```


Confusion Matrix
Actual
Predicted
0 11 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0
1 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
10 0 0 49 0 1 0 0 0 0 0 0 0 0 0 0 2 1 0 0 0 0 0 0 1
11 0 0 0 2 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
12 0 0 0 0 23 2 3 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0
13 0 0 0 0 0 8 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
14 0 0 0 0 0 0 26 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
15 0 0 0 0 0 0 0 20 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0
16 2 0 0 0 0 0 0 0 22 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
17 2 0 0 0 0 0 0 0 0 25 0 0 0 0 0 0 0 0 0 0 0 0 0 0
18 0 0 0 0 0 1 0 0 0 0 6 0 0 0 0 0 0 0 0 2 0 0 0 0
19 0 0 0 0 0 0 0 0 0 0 0 15 0 0 0 0 1 0 0 0 0 0 0 0
2 0 0 0 0 1 0 0 0 0 0 0 0 2 0 0 1 0 0 0 0 0 0 0 0
20 0 0 0 0 0 0 0 1 0 0 1 0 0 15 0 0 0 0 0 0 0 0 1 0
21 2 0 0 0 0 0 0 0 1 0 0 0 0 0 42 2 0 0 1 0 0 0 0 0
23 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 36 0 0 0 0 0 0 0 0
24 0 0 0 1 2 1 1 0 0 0 2 0 1 4 0 0 188 0 4 0 1 0 0 0
3 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 27 0 1 0 0 0 0
4 0 0 0 1 0 0 0 0 0 3 0 0 0 0 0 0 1 0 11 0 0 0 0 0
5 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 109 0 0 0 0
6 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 11 0 0 0
7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 1 0 31 0 0
8 2 0 0 1 5 18 1 3 0 0 10 12 9 35 1 0 14 4 7 17 3 9 297 77
9 0 0 0 0 0 2 0 0 0 0 2 0 0 0 0 0 0 0 0 8 0 0 5 43
0 1 10 11 12 13 14 15 16 17 18 19 2 20 21 23 24 3 4 5 6 7 8 9



Classification Report:

	precision	recall	f1-score	support
0	0.5789	0.8462	0.6875	13
1	1.0000	1.0000	1.0000	5
10	1.0000	0.9074	0.9515	54
11	0.4000	0.6667	0.5000	3
12	0.7188	0.7419	0.7302	31
13	0.2424	0.8889	0.3810	9
14	0.8125	1.0000	0.8966	26
15	0.8333	0.9091	0.8696	22
16	0.9565	0.9167	0.9362	24
17	0.8065	0.9259	0.8621	27
18	0.2609	0.6667	0.3750	9
19	0.5556	0.9375	0.6977	16
2	0.1538	0.5000	0.2353	4
20	0.2778	0.8333	0.4167	18
21	0.9767	0.8750	0.9231	48
23	0.8780	0.9730	0.9231	37
24	0.8952	0.9171	0.9060	205
3	0.8710	0.9310	0.9000	29
4	0.4783	0.6875	0.5641	16
5	0.7842	0.9732	0.8685	112
6	0.7333	0.9167	0.8148	12
7	0.7750	0.9118	0.8378	34
8	0.9706	0.5657	0.7148	525
9	0.3554	0.7167	0.4751	60
accuracy			0.7647	1339
macro avg	0.6798	0.8420	0.7278	1339
weighted avg	0.8535	0.7647	0.7787	1339

```
Last model f-1 score is %76.47 .
```

Future Step

```
    We can work on our model overall shapes to get better result.

    We can gather more images about mars to get better data.(which is a little
problem in this dataset)

    We can try more complex models to get better result.
```

Releases

No releases published

Packages

No packages published

Languages

- **Jupyter Notebook** 100.0%