SUMMER OF INNOVATION

AUBURN WAVES

MISSION MARS TERRAIN

PRANAV KUMAR PANDEY (TEAM LEADER)
MAHESH KRISHNAM (MEMBER 2)

INDIAN INSTITUTE OF TECHNOLOGY, DHARWAD

01

PREPROCESSING STEPS

ML Models:

- Download the Dataset
- Explore and analyze the Dataset
- Prepare the Dataset for ML training
- Train hardcoded and baseline Models
- Make predictions
- Perform feature engineering
- Train and evaluate different models
- Tune hyperparameters for the best suitable Models
- Perform various experimental steps to increase accuracy
- Prediction and evaluation

DL Models:

- Running script to arrange it in format of train -> class folders -> files
- Uploading Dataset to Github
- Downloading Dataset from Github
- Loading dataset using ImageFolder
- Train Validation Split
- Assigning Weight to each Class
- Creating Sampler
- Converting Dataset in DataLoader
- Defining Model
- Setting up required function and matrix
- Training and Testing loop
- Evaluation
- Experimentations
- Prediction

02

MODELS TRAINED

ML Models:

- Linear Regression
- Decision Tree
- Random Forest
- XGBoost

DL Models:

- ANN(Custom Dataset + without weights to each class)
- CNN(Custom Dataset + without weights to each Class)
- CNN(Custom Dataset + weights given to each Class)

TRANSFER LEARNING

Pretrained Models:

- Resnet18
- Resnet152
- Densenet121
- Efficientnet_b0
- Vision Tranformer

DESCRIBING OUR MODELS PROGRESS

STEP 1:

- We were just using ML and Simple ANN models to make predictions on the raw dataset.
- Validation accuracy were low about 50%-60%.
- Training and Validation were not stable.
- Models we are using are not suitable to these kinds of dataset

STEP 2:

- We used Deep CNN models on raw dataset using custom dataset by merging train images and their lables from train.csv.
- We did various experimental steps to decide how can we make our model perform better and better.
- This time we were getting accuracy of somewhere about 80%-85%
- But later realized that we it is misclassifying classes having lesser images.

```
EPOCH : 5
----------
Looked at 0/4960 samples
Looked at 3200/4960 samples
Train Loss : 1.32693 | Train Accuracy : 0.588
Val Loss : 1.29724 | Val Accuracy : 0.596
```


```
EPOCH : 25
----------
Looked at 0.00 samples
Looked at 0.32 samples
Looked at 0.65 samples
Looked at 0.97 samples
Train Loss : 0.47076 | Train Accuracy : 0.836
Val Loss : 0.50457 | Val Accuracy : 0.837

EPOCH : 26
----------
Looked at 0.00 samples
Looked at 0.32 samples
Looked at 0.65 samples
Looked at 0.97 samples
Train Loss : 0.46437 | Train Accuracy : 0.840
Val Loss : 0.49526 | Val Accuracy : 0.835
```

DESCRIBING OUR MODELS PROGRESS

STEP 3:

- Due to high variance in dataset and miss classification on dataset having less images, we introduced some weights to each class.
- Now the CNN with weights give accuracy lesser than the CNN without weights.

```
{'bright dune': 0,
 'crater': 1,
 'dark dune': 2,
 'impact ejecta': 3,
 'other': 4,
 'slope streak': 5,
 'spider': 6,
 'swiss cheese': 7},
```

```
EPOCH : 49
----------
Looked at 0.00 samples
Looked at 0.32 samples
Looked at 0.65 samples
Looked at 0.97 samples
Train Loss : 0.12670 | Train Accuracy : 0.956
Val Loss : 1.53336 | Val Accuracy : 0.731

EPOCH : 50
----------
Looked at 0.00 samples
Looked at 0.32 samples
Looked at 0.65 samples
Looked at 0.97 samples
Train Loss : 0.12756 | Train Accuracy : 0.958
Val Loss : 1.35157 | Val Accuracy : 0.709
Total Time : 371.362 seconds
```

```
{4: 3651, 1: 1062, 0: 597, 5: 335, 7: 223, 2: 216, 6: 66, 3: 51}
```

```
{0: 10.386934673366834, 1: 5.838983050847458, 2: 28.708333333333332, 3: 121.58823529411765, 4: 1.6984387838948234,
5: 18.51044776119403, 6: 93.95454545454545, 7: 27.807174887892376}
```

- And also validation was little bit unstable.

DESCRIBING OUR MODELS PROGRESS

STEP 4:

- Due to all these issues we moved towards pretrained models such as resnet, densenet and efficiencynet.
- There we were getting accuracy more than 90% with weights on class and a stable training.

STEP 5:

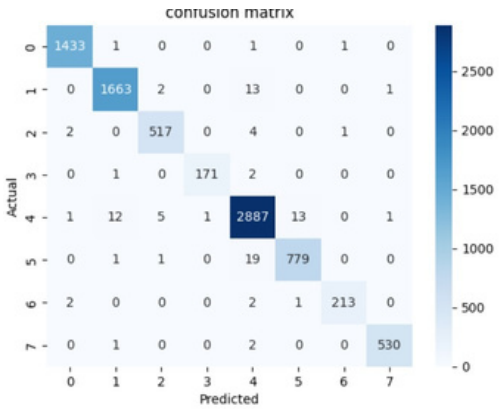
- We finalized our pretrained model name resnet152 with accuracy of about 94% on val dataset.
- We merged the train and validation split and finally fine tuned the resnet152 on whole dataset.
- Finally used this model to make prediction over test dataset and stored inside mission_mars_terrain_submission.csv

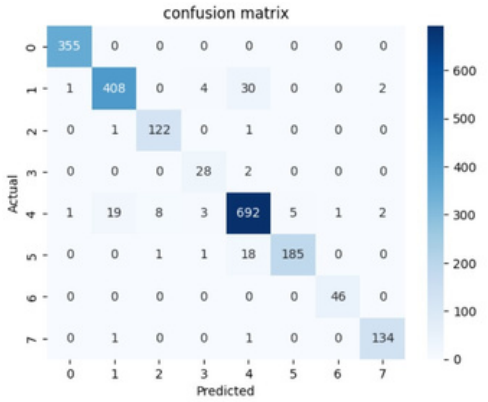
05

DESCRIBING OUR MODELS PROGRESS

STEP 5:

```
Epoch 17/40, Train Loss: 0.0586, Train Accuracy: 97.96%, Val Loss: 0.2017, Val Accuracy: 93.53%
/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:558: UserWarning: This DataLoader wil
  warnings.warn(_create_warning_msg(
Epoch 18/40, Train Loss: 0.0335, Train Accuracy: 98.89%, Val Loss: 0.2842, Val Accuracy: 92.81%
/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:558: UserWarning: This DataLoader wil
  warnings.warn(_create_warning_msg(
Epoch 19/40, Train Loss: 0.0623, Train Accuracy: 97.88%, Val Loss: 0.2701, Val Accuracy: 93.24%
/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:558: UserWarning: This DataLoader wil
  warnings.warn(_create_warning_msg(
Epoch 20/40, Train Loss: 0.0413, Train Accuracy: 98.52%, Val Loss: 0.2426, Val Accuracy: 93.68%
```





THANK YOU !!

MISSION MARS TERRAIN