

Final Project Proposal

1. Project Description

- The final project chosen for MAIS202 is a Planetary Surface Anomaly detector specifically for rovers on Mars. The aim of the project is object classification of objects on Mars.

2. Dataset

- [Kaggle / Nasa Curiosity Rover Data Collection](#), if needed, images from [NASA's Planetary Data System](#) might be scraped.
- The datasets contain real Martian surface photographs with varied lighting and terrain, which is ideal for training robust visual models. The dataset in Kaggle has many helpful guidelines such as the code section and existing community notebooks with examples.

3. Methodology

- a. Data Preprocessing: To reduce domain variability, we can pick a single camera type from the rover (Mastcam, DESCAM, HAZCAM etc...). The images will be resized, normalised and cleaned from corrupted files if there are any. 5-7 object classes (sedimentary rock, soil, rover_part, shadow, volcano) and manually label a seed set (~200-500 images per class). Train/validation/test split will be stratified (70/15/15).
- b. Machine Learning Model: We will use transfer learning with a pretrained CNN like ResNet and replace the final classifier for the Mars object classes and easy to deploy in webapps. Transfer learning is chosen for fast convergence and string performance on small datasets. If time permits with box labeling of the images, we will prototype an object-detection model like YOLOv8 as an extension.
- c. Evaluation Metric: For classification, accuracy, per-class precision, recall and confusion matrix. If we add detection, mean Average Precision. The baseline to beat: simple pretrained model with no fine-tuning and random classifier. Target to achieve >75% per-class F1 on common classes.

4. Application:

- We will build a simple webapp with Flask + React where a user uploads a rover image and receives predicted object label(s) with confidence scores. If detection is implemented, the app will display bounding boxes. For transparency, we could include highlighting important regions. The app will be hosted for the demo; model inference will run server-side.

