Lunar Crater and Boulder Detection using CNN-based Object Detection

Team Box Box Engineers

Team Members: Affan Pathan, Shreyas Reddy, Sreejita Chatterjee

Problem Statement

The objective of this project is to develop an AI/ML-based system, preferably using Convolutional Neural Networks (CNNs), to detect and localize craters and boulders of various shapes and sizes from high-resolution lunar surface imagery. The system must generate bounding boxes around detected features while being robust to varying scales, lighting conditions, and terrain complexities. An optional goal is to design a web-based interface for image upload and detection visualization.

Approach

Our approach evolved through two primary phases: an initial baseline using YOLO-Nano, followed by the design of a custom lightweight detection model.

1. YOLO-Nano Baseline

We trained a YOLO-Nano model on approximately 4,000 labeled samples from the provided 13,000-image dataset for about 100 epochs. Despite limited compute, the model achieved a detection accuracy of around 75%, and demonstrated good generalization to unseen images. It formed the core of our detection pipeline.

2. Custom Model Architecture

While YOLO-Nano was effective, we recognized that crater and boulder detection on lunar imagery did not require the complexity of a full-scale YOLO model. Hence, we developed a lightweight custom CNN-based detection architecture inspired by YOLO principles.

The model comprises:

- A backbone of 5 convolutional blocks with increasing channel depth (32 to 512), each consisting of convolution, batch normalization, LeakyReLU activation, and max pooling.
- A detection head that processes feature maps and outputs bounding box predictions and confidence scores.
- Architecture is optimized for single-channel grayscale imagery, aligning with lunar image characteristics.
- We designed a lightweight model, only **1.57 million parameters**, significantly lighter that compared to YOLOv8 Nano (**3.16 million params**) or YOLOv8 Small (**11.6 million params**) if trained long enough.

Loss Function Design

Our custom loss function builds upon the YOLO paradigm, with enhancements to improve training stability and localization performance:

- **CIoU-based Bounding Box Loss:** Captures overlap, distance, and aspect ratio discrepancies between predicted and ground-truth boxes.
- **Objectness Loss:** Penalizes incorrect object confidence predictions using binary cross-entropy.
- **Spatially-Aware No-object Loss:** Instead of uniformly penalizing all negative detections, we apply a distance-weighted penalty—focusing more on spatial proximity to actual objects and less on distant cells.

This formulation aims to improve training efficiency and detection quality, especially under sparse object distribution as seen in lunar imagery.

Challenges and Limitations

- **Training from Scratch:** Our custom model lacked pre-trained weights and required long training cycles (800+ epochs) to converge effectively.
- **Hardware Constraints:** Without access to TPUs or multi-GPU clusters, training times were prohibitively long, restricting experimentation.
- **Large Dataset:** Managing a high-resolution 13k-image dataset presented challenges in preprocessing, storage, and training throughput.

Final Strategy

Given time and hardware constraints, we chose to proceed with the YOLO-Nano model for our final submission. It performed reliably and allowed us to build a complete inference and UI pipeline. However, we have included our custom model in the submission repository for evaluation and future benchmarking.

Bonus Task: Web-based User Interface

To enhance usability, we implemented a browser-based interface using **Streamlit**, allowing users to:

- Upload lunar surface images for real-time analysis.
- Visualize bounding box detections and predicted object locations.
- Generate a hazard map identifying risky and safe zones for rover navigation.

Bonus task: Terrain Analysis and Autonomous Rover Path Planning System for Lunar Surface Exploration

This module introduces a system for **lunar terrain analysis** and **autonomous rover navigation**, aimed at enhancing mission planning and operational safety. The approach integrates computer vision and pathfinding algorithms to simulate real-world utility for robotic lunar exploration. The key components of the system are:

- **Crater Detection:** Utilizes a YOLOv8-based object detection model trained to identify lunar craters in visual input data.
- **Edge-Aware Preprocessing:** Applies Canny edge detection to highlight terrain discontinuities, which are then blended with the original image for enhanced feature representation.
- **Hazard Map Generation:** Constructs a risk map by combining detected crater regions and edge intensity information, which reflects hazardous areas on the lunar surface.
- **Simulated Depth Estimation:** Generates a pseudo-depth map from edge data to infer relative terrain elevations, aiding spatial awareness.
- **A* Pathfinding Algorithm:** Computes an optimal path between a start and goal location while avoiding high-risk zones identified in the hazard map.

This modular pipeline facilitates safe and efficient navigation route planning for robotic rovers, providing practical value in future lunar exploration missions.

Conclusion

Our project combines the robustness of established object detection frameworks with the exploration of a custom-tailored lightweight solution. While we encountered practical limitations during development, we successfully delivered an accurate, scalable system complemented by a user-friendly interface.

We further enhanced interpretability through hazard heatmaps and confidence histograms, and improved training efficiency using AMP and progressive resizing. These upgrades collectively support reliable and resource-aware deployment for real-world planetary missions.