# Rockfall Detection on Lunar Images

## Overview

This project focuses on detecting rockfalls on the Moon by analyzing lunar surface images. The goal is to identify craters formed by rockfall events and provide precise bounding box coordinates for each detected crater. The model combines convolutional neural networks (CNNs) with image processing techniques to localize craters in both positive images (with craters) and negative images (without craters).

## Key Features

1. Rockfall Identification: Detects rockfall events by locating craters in lunar images, using bounding box annotations to represent each crater's position.  
2. Bounding Box Prediction: Provides accurate bounding boxes around detected craters and ensures no bounding boxes are returned for negative images.  
3. Image Normalization and Padding: Prepares images of different sizes by padding them to a uniform dimension, allowing efficient model processing.  
4. Multiple Bounding Box Handling: Capable of processing images with multiple craters by using up to 10 bounding boxes per image.

### Technologies Used

- Deep Learning: TensorFlow and Keras for CNN model development and training.  
- Image Processing: OpenCV for loading and pre-processing images, including resizing and normalizing bounding box coordinates.  
- Data Handling: NumPy and Pandas for data management and label handling.  
- Training and Evaluation: Model training and validation with metrics like accuracy and mean squared error for bounding box prediction.

## System Architecture

- Data Preprocessing: Loads and resizes images to a 256x256 square format, adjusts bounding boxes, and normalizes coordinates. Both positive and negative samples are processed for model input.  
- Model Training: The model is trained on a custom CNN architecture with residual blocks to enhance learning efficiency and robustness against overfitting.  
- Bounding Box Prediction: After training, the model outputs bounding boxes for detected craters and ignores regions in images labeled as 'no rockfall'.  
- Visualization: Bounding boxes are drawn on images during evaluation to visualize detection accuracy.

## How It Works

1. Data Loading: Reads images and labels, with labels including bounding box coordinates for craters. Negative images are labeled with bounding box parameters set to (-1, -1, -1, -1).  
2. Image Processing: Pads and resizes images, and normalizes bounding box coordinates to align with the model’s expected input size.  
3. Model Training: The model trains over 8 epochs with high accuracy achieved for classification and bounding box regression tasks.  
4. Prediction and Visualization: The model is evaluated on test images, and bounding boxes are visualized to demonstrate the model’s accuracy in localizing craters.

## Functionalities

1. Bounding Box Detection: Predicts bounding boxes for craters in positive images.  
2. Negative Image Handling: No bounding boxes are returned for negative images, aligning with the task's objective.  
3. Crater Localization: Visualizes bounding boxes on images to showcase precise crater locations.

### Issues

- Bounding Box Accuracy: Some bounding boxes may slightly misalign with craters due to image resizing and padding adjustments.  
- Negative Image Detection: A few false positives in negative images, likely due to slight feature similarity with craters.  
- Computation Time: Training on high-resolution lunar images requires significant computational resources.

## Future Improvements

1. Model Optimization: Use transfer learning or a more complex architecture for improved accuracy.  
2. Additional Data: Increasing the dataset size with more annotated lunar images may enhance generalization.  
3. Automated Labeling: Implement semi-supervised methods to handle data annotation efficiently.  
4. Error Analysis: Analyze and refine bounding box predictions to further reduce errors.