Lunar Crater Detection Using YOLOv8

Submission for

IIT Dharwad: Summer of Innovation, 2025

Raghuveer Verma

June 18, 2025

GitHub Repository

1 Approach

In this project, a YOLOv8 object detection model was trained to detect craters on the lunar surface. Due to hardware constraints on the local machine, the **YOLOv8n (nano)** variant was selected, which is optimized for low-resource environments. Despite being lightweight, YOLOv8n provided reasonably good detection performance.

2 Architecture

Standard YOLOv8 architecture showed the best results.

3 Experimentation

The model was trained for 55 epochs on 13,956 images and validated on 3,741 images [dataset available here].

In a later experiment, it was attempted to ensemble the predictions of the two best-trained models using majority voting on averaged bounding-box coordinates, confidence score, and class [code available here].

However, this strategy led to a significant drop in recall, which is a critical metric for this application [see Figure 1].

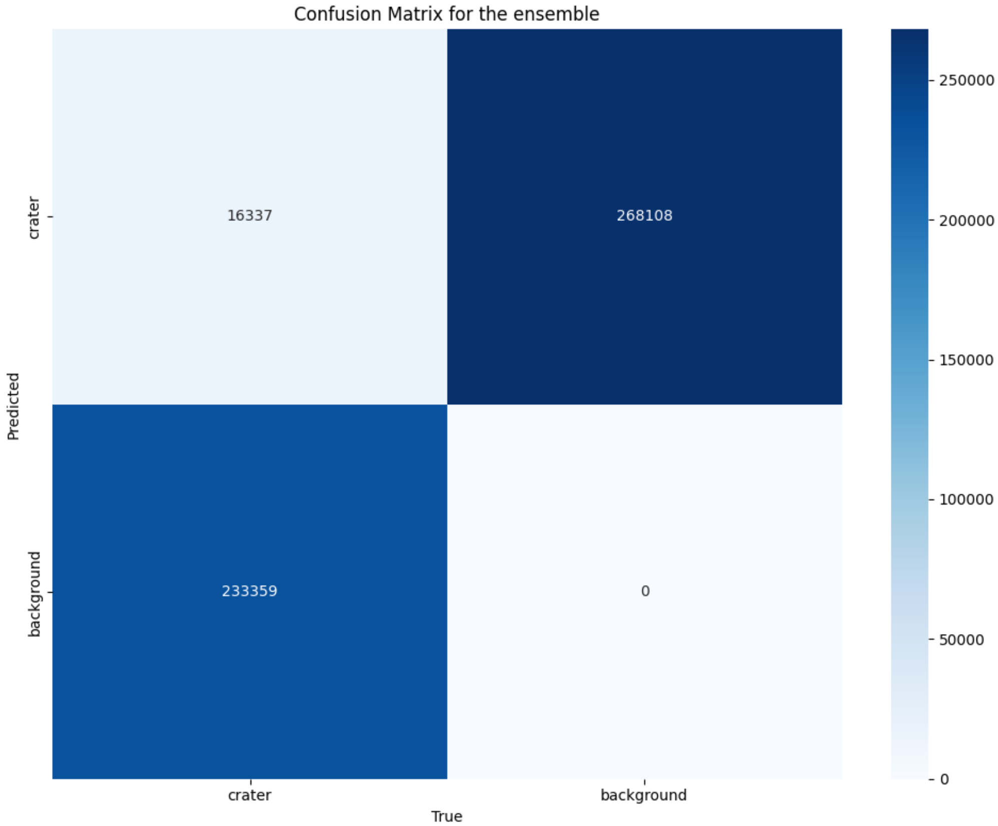

Figure 1: Confusion Matrix for the ensemble

4 Challenges Faced

- **Limited Training Epochs:** Due to RAM limitations, the model was not trained for a large number of epochs which could potentially have improved performance.
- **GPU Configuration:** The system used has an Intel Iris Xe GPU, but attempts to exploit it were unsuccessful due to limited system resources such as disk space to install the required libraries.
- **Inconsistent Labeling:** The provided data has inconsistent labeling and some craters which are not labeled. This affected the training adversely and the model is unable to detect smaller craters.

5 Performance on Validation Set

The best-performing model achieved the following metrics on the validation set:

- **Recall:** 85.40% @ `conf=0.23`
- **mAP50:** 86.29%
- **mAP50-95:** 70.20%

Since the goal is to detect **all possible craters**, recall is prioritized over precision. A higher false-positive rate is acceptable because the model often detects craters that are actually present but not labeled in the dataset [see Figure 4].

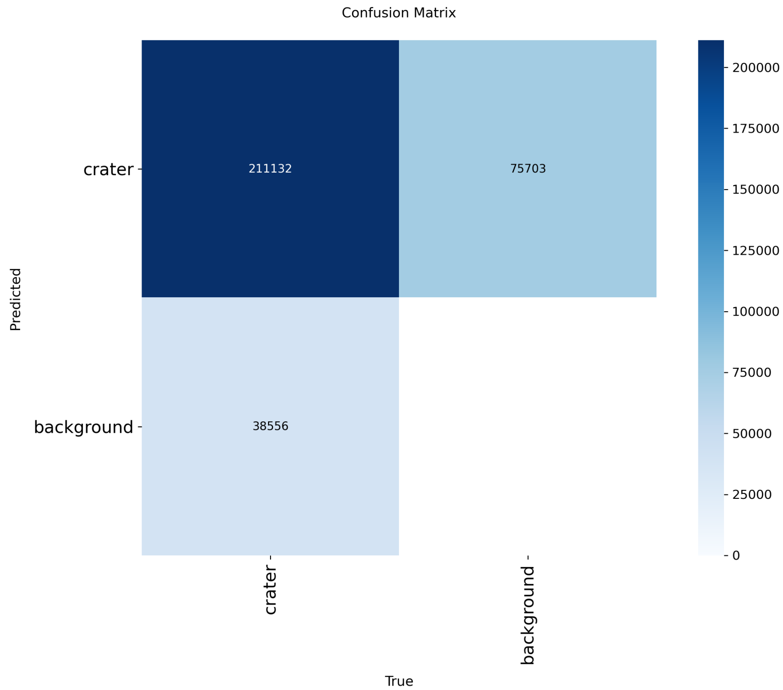

Figure 2: Confusion Matrix for the best performing model with default parameters on the validation set

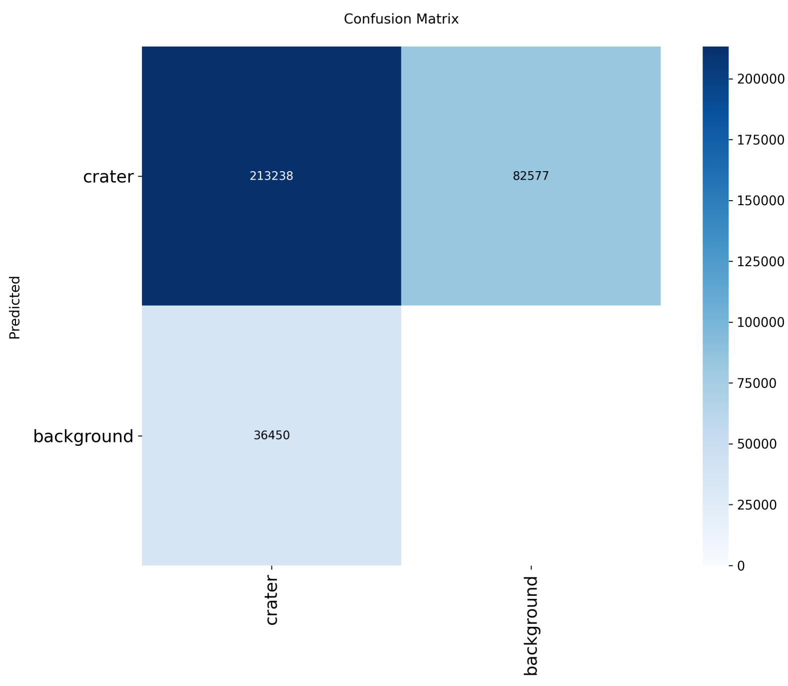

Figure 3: Confusion Matrix for the best performing model @ `conf=0.23` on the validation set

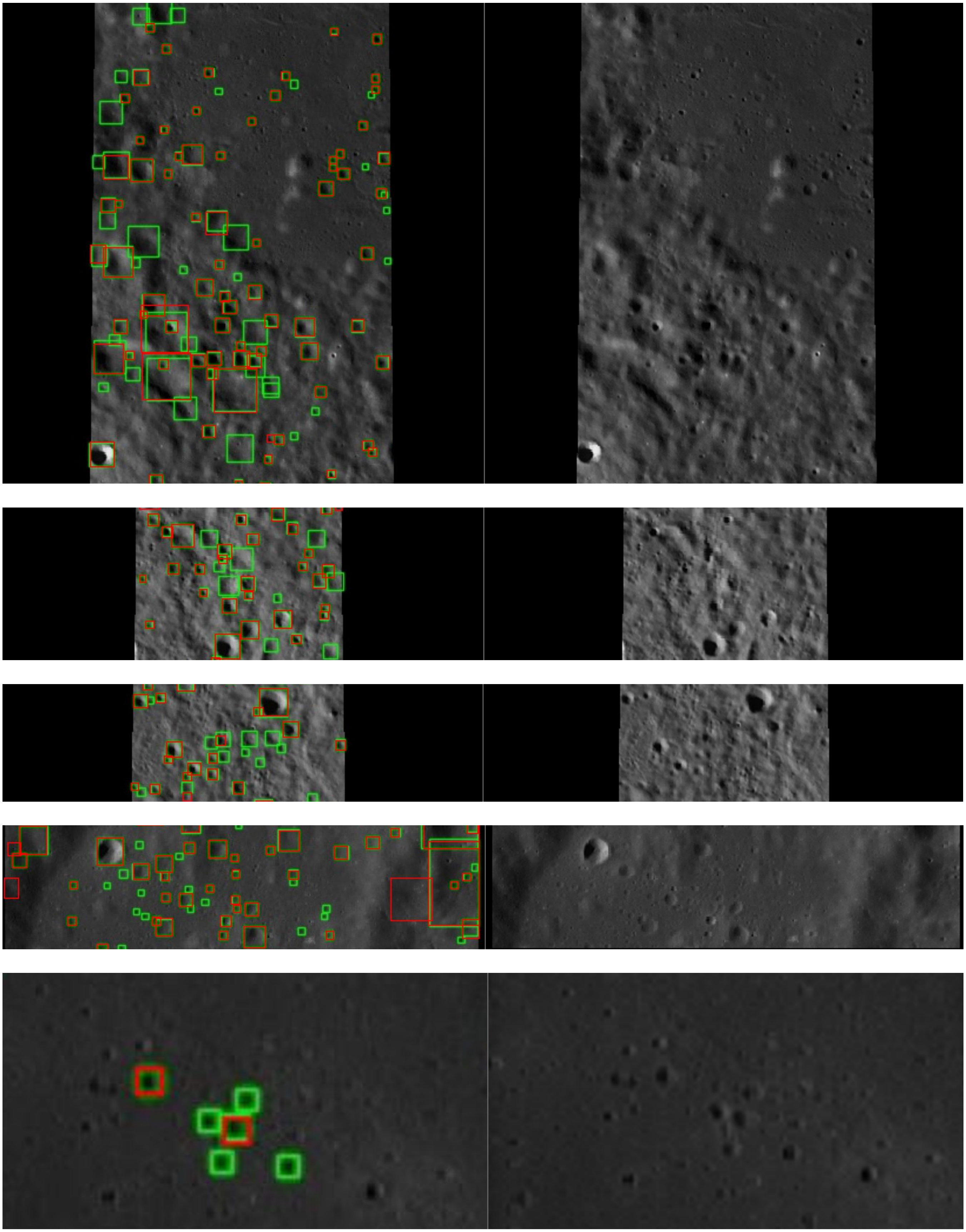

Figure 4: Left: Images with labeled bounding boxes (red) and model predictions (green); Right: Original Images. The original images, labels and predictions are available here.

6 UI Implementation

To make the crater detection process accessible, a user-friendly **Flask-based web application** was built [link] with the following features:

- **Image Upload:** Users can upload lunar surface images and receive annotated images with bounding boxes for detected craters.
- **Download Results:** The bounding box data is downloadable in YOLO format as a `.txt` file.
- **Live Detection:** The app supports real-time crater detection using the system's webcam.

This functionality can be extended in the future to assist autonomous lunar landing systems by detecting craters in real-time and guiding landers toward safer zones.

7 Conclusion

Despite limited computational resources, YOLOv8n provided a viable solution for the detection of lunar craters. With further labeling, better hardware, and additional training, the model's accuracy and usability in real-world space missions can be significantly improved.