Robot Object Detection

1. Introduction

Objective:

- Develop an object detection model for identifying and localizing sample tubes on Mars for the Sample Fetch Rover.

Motivation:

- Space exploration rovers like Perseverance leave behind sample tubes for future missions.
- Detecting these tubes efficiently with low-resource models supports mission goals while reducing computational and power constraints.

Dataset:

- Combines real images from the Planetary Utilisation Testbed (PUT) and synthetic images generated with photorealistic simulators.
- Includes grayscale transformations for Mars rover camera compatibility.

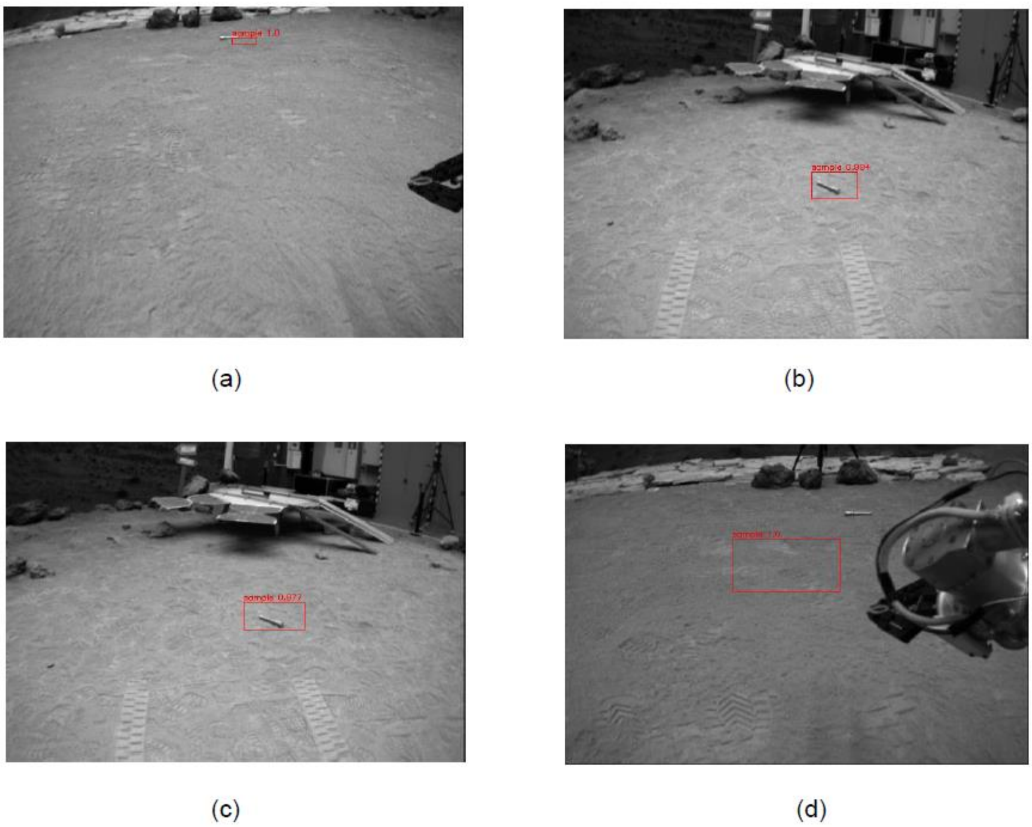


Figure 1 - Model Prediction Results with the Test Set, Demonstrating Bounding Box Predictions on Sample Images.

2. Methodology

Model Design:

- Two Convolutional Neural Networks (CNNs):
  - **Detection Model**: Classifies whether an image contains a sample.
  - **Regression Model**: Predicts bounding box coordinates for the detected sample.

Optimization Techniques:

- Used Adam optimizer and dynamic learning rate schedules.
- Employed data augmentation with Albumentations.ai for robustness.

Model Constraints:

- Designed for low RAM usage and power efficiency.
- Limited convolutional layers and kernel sizes to ensure lightweight architecture.

3. Results

Model Performance:

- **Detection Model**:
    - Precision and recall values evaluated using confusion matrix metrics.
- **Regression Model**:
    - Bounding box predictions assessed using Intersection over Union (IoU).

Comparative Analysis:

- Compared with YOLOv8:
    - YOLO achieved significantly higher accuracy and precision.
    - Custom model showed poor results due to dataset limitations and computational constraints.

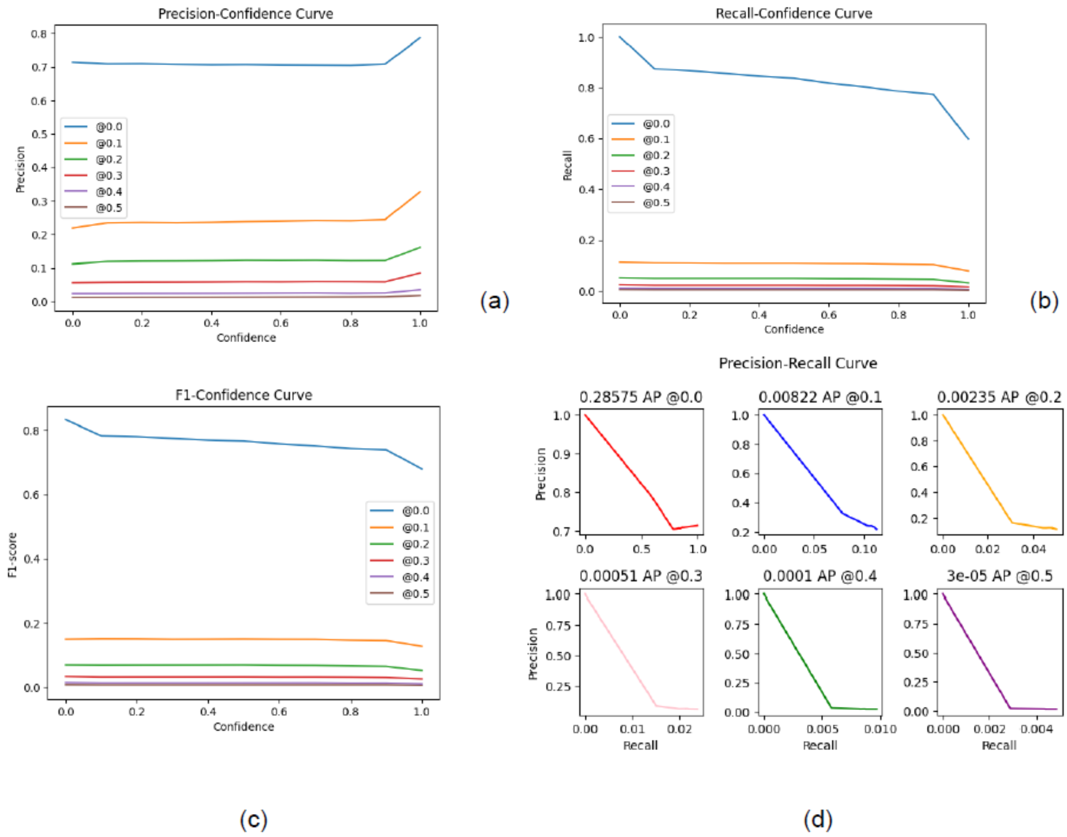
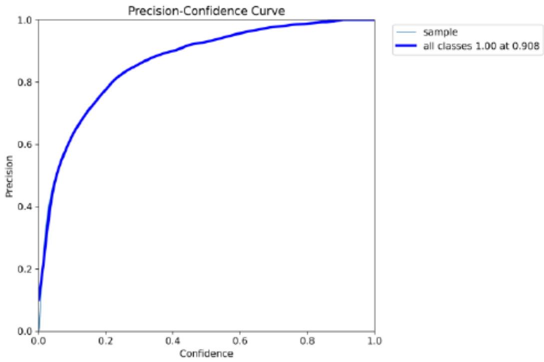
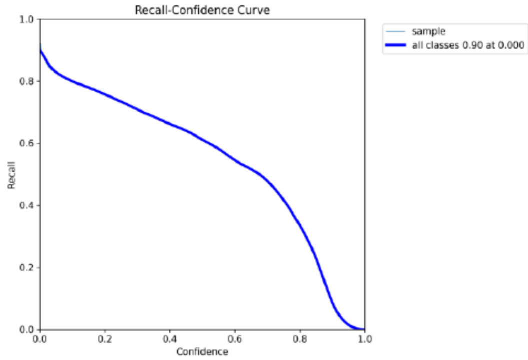


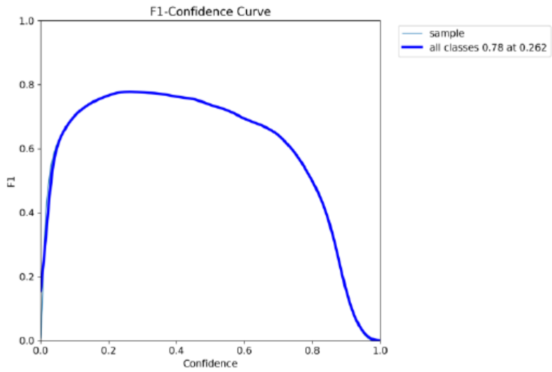
Figure 2 - Precision-Confidence (a), Recall-Confidence (b), F1-Confidence (c), and Prediction-Recall (d) Curves of the Custom Detection Model, Showing Performance Metrics under Different IoU Thresholds.



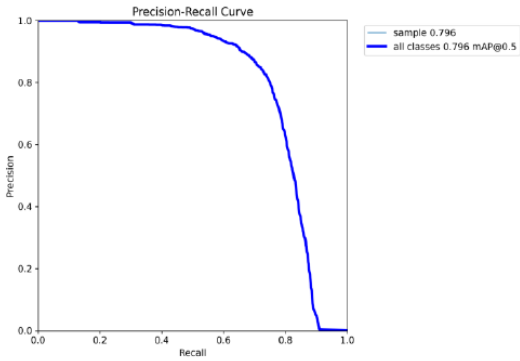
(a)



(b)



(c)



(d)

Figure 3 - YOLO Precision-Confidence (a), Recall-Confidence (b), F1-Confidence (c) and Prediction-Recall (d) Curves for Comparison with the Custom Model.

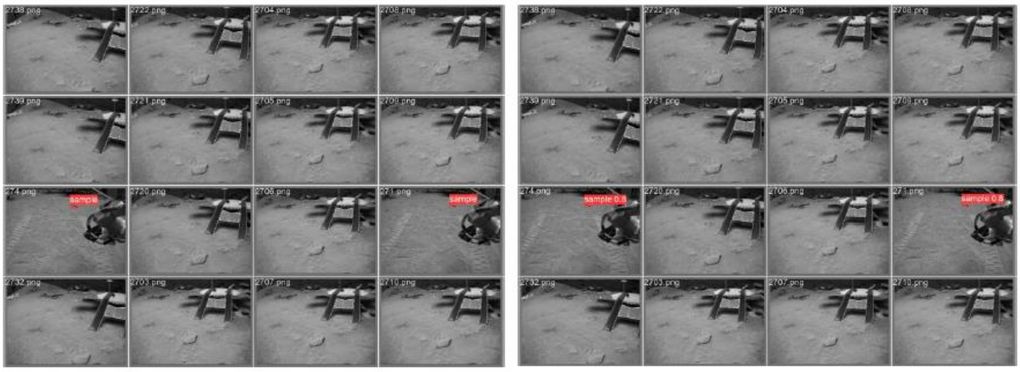
4. Discussion

Challenges:

- Bias is introduced by synthetic images lacking realistic shadows and details.
- Grayscale conversion and resizing reduced sample visibility.

Insights:

- YOLO's pre-trained architecture performed better due to extensive training and data diversity.
- Custom models require further refinement and computational resources for significant improvements.



(a) (b)

Figure 4 - YOLO Prediction Results Compared with Ground Truth Bounding Boxes, Illustrating the Advantages of YOLO's Detection Accuracy.

```
- for y_test_detection and y_pred_detection:
mean % of the area intersected in all intersections: 14.93403693931398
number of intersections: 379
percentage of intersections: 13.14147018030513
```

Figure 5 - Summary of Intersection Metrics, Including Mean Area, Number and Percentage of Intersections Between Predicted and Ground Truth Bounding Boxes.

5. Conclusion

Strengths:

- Successfully developed two lightweight CNNs for object detection and bounding box regression, addressing the challenge of creating models suitable for low-power applications.
- Demonstrated the feasibility of using grayscale and resized images to align with the constraints of Mars rover hardware.
- Provided a foundation for exploring object detection in resource-constrained environments, highlighting the potential of lightweight models for space exploration tasks.

Areas for Improvement:

- Detection Accuracy: The custom model exhibited significantly lower performance compared to pre-trained YOLOv8, with IoU and precision metrics well below desired thresholds. This highlights the need for improved dataset quality and model architecture.
- Synthetic Dataset Bias: Training on a dataset heavily reliant on synthetic images introduced biases, particularly due to unrealistic shadows and object placement.
- Computational Resources: Limited RAM and hardware constraints impeded model training and experimentation, necessitating design compromises that affected overall performance.
- Image Processing: Grayscale transformations and resizing reduced sample visibility, contributing to the poor detection of smaller objects.