Space Station Safety Equipment Detection Challenge — Final Hackathon Report

Team Information

Team Name: InnovexHQ
Project Name: Space Station Safety Equipment Detection
Tagline: Ensuring Astronaut Safety Through Automated Equipment Monitoring

Team Members

Tarun SM – Model Engineering and Optimization
Tarun led the development and refinement of the YOLO-based detection model. He ensured that the hyperparameters, augmentation strategies, and training pipeline were optimized for accuracy and stability. His expertise in fine-tuning and debugging complex deep learning workflows played a central role in the final model performance.

Diwakar S – Dataset Preparation, Evaluation, and Integration
Diwakar managed the complete dataset pipeline, including preprocessing, label structuring, augmentation, and validation setup. He also handled model evaluation, performance analysis, and integration with the real-time inference system. His strong focus on clarity and reproducibility helped shape the documentation and overall project organization.

Venkatesh Shivanand Kolabal – Real-Time Detection and Tracking
Venkatesh worked on the real-time deployment of the model, addressing issues such as bounding box flickering, temporal instability, and performance drops during live detection. He implemented tracking algorithms, confidence threshold tuning, and frame-skipping strategies that made the model practical for real-world use.

Varun J – Code Structure, Error Handling, and System Stability
Varun designed and structured the modular codebase for easier testing and reproducibility. His contribution included writing clean, maintainable scripts for training, inference, and utility functions. He added robust error-handling mechanisms, ensuring that the system remained stable even in low-memory or noisy environments.

Sunil K – Documentation, Presentation, and Workflow Organization
Sunil focused on the documentation, ensuring that the project remained understandable, professional, and easy to reproduce. His work included preparing structured reports, organizing results, and maintaining presentation-ready visualizations such as training curves, confusion matrices, and failure case analyses.

1. Introduction

Safety-critical equipment such as fire extinguishers, oxygen tanks, and emergency communication panels are essential for astronaut survival in space station environments. Manually verifying the availability and correct placement of this equipment is time-consuming and unreliable. Our project addresses this challenge by building an automated object-detection system that monitors seven essential safety items using deep learning.

2. Methodology

2.1 Dataset Preparation

The dataset provided for the challenge consisted of synthetic images representing various space-station environments. It was organized into the following splits:

- Training: hackathon2_train_1/train_1/
- Validation: hackathon2_train_1/val_1/
- Testing: hackathon2_test3/test3/ (1,408 images)

The system was trained to detect seven classes of safety equipment:

1. OxygenTank
2. NitrogenTank
3. FirstAidBox
4. FireAlarm
5. SafetySwitchPanel
6. EmergencyPhone
7. FireExtinguisher

All images were preprocessed to 640×640 resolution. Label files were converted to YOLO format, and data augmentation included flips, scaling, color variations, and mosaic augmentation.

2.2 Model Training Approach

We selected YOLOv8-Small due to its balance between accuracy and speed.

Training Configuration

Parameter	Value
Epochs	50
Initial Learning Rate	0.001
Final Learning Rate	0.01
Momentum	0.937
Batch Size	16
Image Size	640
Mosaic Probability	0.5
Early Stopping	50 patience

Key Training Enhancements

- Gradual learning-rate scheduling
- Strong augmentation to overcome synthetic data limitations
- Validation-based early stopping to prevent overfitting
- Monitoring of box loss, classification loss, and mAP during training

2.3 Evaluation Method

The evaluation strategy included:

- Overall mAP@0.5 and mAP@0.5:0.95
- Precision-Recall per class
- Confusion matrix analysis
- Annotated sample outputs from the test set
- Performance on real-time webcam input

Testing covered all 1,408 images in the test set.

3. Results and Performance Metrics

3.1 Overall Model Perf

Metric	Metric
mAP@0.5	mAP@0.5
mAP@0.5:0.95	mAP@0.5:0.95
Precision	Precision
Recall	Recall

The model achieved moderate accuracy given the challenges of synthetic data, visually similar classes, and varied lighting conditions.

Class	mAP@0.5	Precision	Recall
OxygenTank	0.496	0.73	0.414
FirstAidBox	0.458	0.583	0.431
FireExtinguisher	0.262	0.45	0.321
NitrogenTank	0.229	0.495	0.221
FireAlarm	0.178	0.362	0.155
SafetySwitchPanel	0.14	0.23	0.218
EmergencyPhone	0.107	0.183	0.221

3.2 Class-Wise Performance

Best detection performance was achieved for OxygenTank and FirstAidBox. The lowest scores were observed for EmergencyPhone and SafetySwitchPanel, likely due to fewer samples and higher visual overlap with other objects.

3.3 Confusion Matrix Insights

From the analysis:

- OxygenTank and FirstAidBox showed consistent detection.
- NitrogenTank and OxygenTank were often confused with one another.
- EmergencyPhone and SafetySwitchPanel had high misclassification rates.
- Small objects and those captured under poor lighting conditions were often missed.

4. Challenges and Solutions

4.1 Memory Management Limitations

Large images during augmentation caused memory overflow.

Solution:

- Automatic image resizing
- Controlled batch sizes
- Efficient data loading
- Fallback systems to avoid crashes

4.2 Flickering Issues in Real-Time Detection

Bounding boxes fluctuated heavily during webcam usage.

Solution:

- Added IOU-based frame-to-frame object tracking
- Reduced confidence threshold to 0.25
- Processed every third frame to stabilize motion
- Implemented temporal consistency checks

4.3 Poor Performance on Certain Classes

EmergencyPhone and SafetySwitchPanel had low detection accuracy.

Solution:

- Analyzed dataset imbalance
- Applied targeted augmentation for under-represented classes
- Proposed hybrid training with real-world samples
- Suggested domain adaptation for synthetic-to-real improvements

5. Optimizations

5.1 Training Optimizations

- Extended training duration
- Tuned learning rate and momentum
- Improved augmentation pipeline

5.2 Inference Optimizations

- Frame skipping for real-time detection
- Lower image resolution for faster inference
- Model warm-up to ensure consistent performance

5.3 Code Structure Enhancements

- Modular directory structure
- A unified main script for training and inference

- Comprehensive error-handling and logging
- Clean separation of model, dataset, and utility files

6. Performance Evaluation

6.1 Training Curves and Trends

The training progress showed:

- A steady decline in classification and box loss
- Improved stability in the validation set
- No signs of overfitting due to early stopping
- Smooth convergence after 40 epochs

6.2 Failure Case Analysis

Key patterns included:

- Partial occlusion leading to low confidence scores
- Extreme lighting (overexposure or shadows)
- Visually similar objects causing cross-class confusion
- Smaller objects often going undetected

6.3 Observations

- Distinct, well-lit objects performed best
- Class imbalance significantly affected recall
- Synthetic environments did not fully replicate real-world complexity

7. Conclusion and Future Work

7.1 Project Summary

We developed a YOLO-based detection system capable of recognizing seven safety-critical objects in a space-station environment. With an mAP@0.5 of 0.2672, the model performed reliably in clear conditions and showed potential for further development.

7.2 Accomplishments

- Built a functional and reproducible pipeline
- Achieved real-time detection with reduced flickering
- Designed a modular, scalable code structure

- Generated comprehensive documentation and visual evidence

7.3 Future Improvements

- Collect more diverse real-world training data
- Use larger models such as YOLOv8-Medium or YOLOv9
- Balance the training dataset across all classes
- Explore ensemble-based detection approaches
- Apply active learning for continuous model refinement

7.4 Impact

This solution lays the foundation for an automated monitoring system within a space station. With further development, it could significantly reduce manual inspection time, improve safety readiness, and support mission-critical operations in harsh environments.