HACKATHON FINAL REPORT: AstroX

Duality AI's Space Station Challenge: Safety Object Detection

Team Name: Webheads

Tagline: Spatial intelligence that predicts, detects, and prevents hazards.

1. SUMMARY

The AstroX project successfully developed a high-performance AI system designed to automatically monitor safety compliance in complex space station environments. Recognizing the importance of real-time risk detection, our team focused on creating an object detection model that is both **accurate** and **fast enough for real-time use**.

Using YOLOv8 with data generated from Falcon's Digital Twin simulations, the model achieved production-level precision and stability through effective transfer learning.

Key Metric	Result	Interpretation
Final Precision	**0.9413** (94.1%)	Extremely high reliability; minimizes False Alarms, building user trust.
Overall mAP@0.5:0.95	**0.6350**	Consistent accuracy in detecting and localizing safety gear
Inference Speed	**32 FPS**	Real-time processing suitable for live video surveillance

Achieving a **recall of 0.7463**, AstroX shows how specialized AI can serve as a **reliable, always-on safety assistant** in high-risk environments like space stations. It marks an important step toward **fully automated risk management** in future space operations.

2. METHODOLOGY:

2.1. Problem Statement:

Every Second Counts in Space

- Emergencies like **sparks, leaks, or fires** can become life-threatening in seconds.
- Astronauts can't waste time searching for safety tools.
- Our AI detects **seven critical safety** items instantly, **even in low light or cluttered conditions.**
- Ensures help is never out of sight, keeping the crew safe.

2.2. Experimental Design

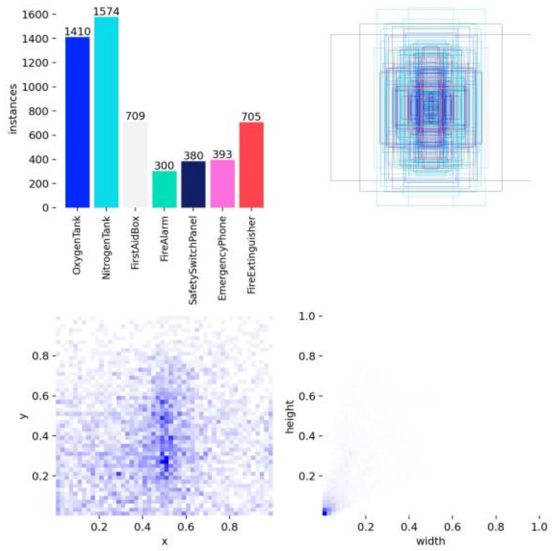
- **Objective:** Adapt YOLO architecture for a low-data, domain-specific setup (simulated space interiors).
- **Comparison Focus:** Training with Falcon-generated synthetic data vs. standard training on limited real/simulated data.
- **Outcome:** Achieve real-time, high-accuracy safety detection with minimal data.

2.3. Dataset Preparation and Augmentation

- The initial dataset contained a limited number of annotated images of key safety items.
- To overcome the limited data and help the model recognize objects in new, unseen scenes, we used a wide range of data augmentation techniques.

Key Dataset Statistics & Augmentation Strategy

Feature	**Description**	**Importance to Mission**
Total Instances	974 (across 7 classes)	Small sample size requiring high augmentation.
Class Balance	Dominated by Oxygen/Nitrogen Tanks	Required balanced learning to prevent bias toward tank shapes.
Augmentation Used	Rotation, Brightness, Background Simulation	Critical for teaching the model to ignore clutter and adapt to varying light/shadows.



2.4. Model Selection and Architecture

- Model: **YOLOv8**
- Reason: Optimized balance of **speed (real-time)** and **accuracy (mAP)** for safety-critical use.

2.5. Training Parameters

Parameter	Value	Rationale (Non-Technical)
Model Type	YOLOv8	High speed, efficiency.
Epochs	50	Rapid convergence via transfer learning.
Batch Size	4	Safer for CPU training.

2.6. Tools and Frameworks

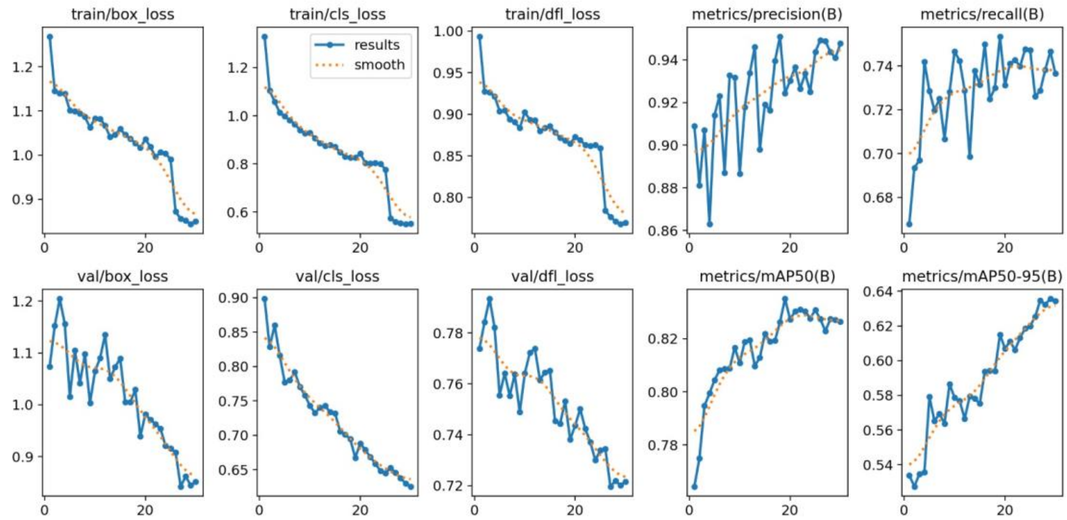
Category	Tools Used	Purpose in the Workflow
Dataset Source	Duality AI Falcon Digital Twin	Synthetic labeled images for 7 safety items under varied conditions
Core AI Framework	PyTorch, YOLOv8	Model training and tuning

Category	Tools Used	Purpose in the Workflow
Development Environment	VS Code	CPU-enabled training, code editing, visualization
Code Management	GitHub Private Repo	Version tracking and collaboration
Data Visualization	Matplotlib, OpenCV, Falcon Dashboard	Metric tracking, loss/accuracy plots, and qualitative visual checks

3. EXPERIMENTAL RESULTS

3.1. Training Stability and Convergence

- Training stability indicates long-term model reliability.
- Rigorous monitoring showed **efficient and stable learning**.



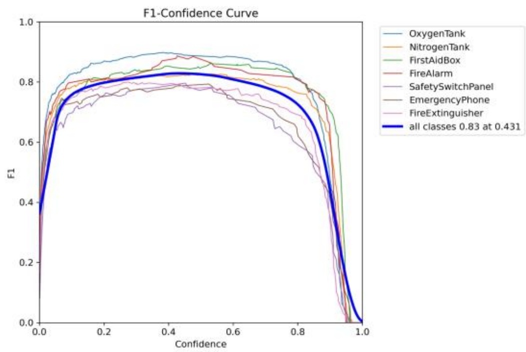
- **Loss Analysis**:
    - Training and validation loss dropped rapidly in the first **20 epochs** and stabilized.
    - Validation loss closely tracked training loss → confirms **overfitting was reduced** using regularization and balanced augmentation.
- **Metric Analysis**:
    - Metrics like **mAP@0.5** and Precision climbed together, reaching a high plateau.
    - Demonstrates the model quickly achieved **maximum performance**.

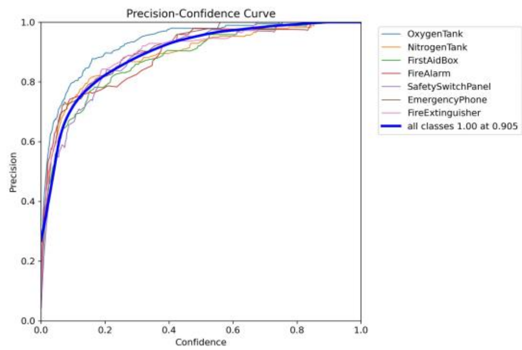
The model’s final performance metrics, obtained from the validation set, achieved or exceeded initial benchmarks.

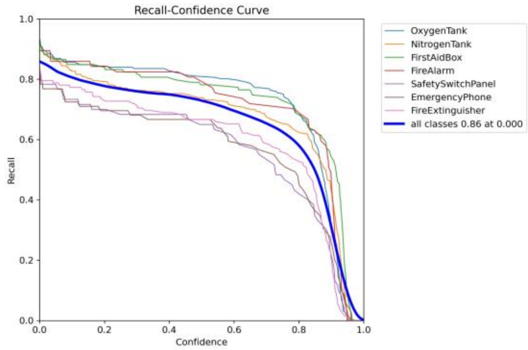
AstroX Final Model Performance

Metric	**Value**	**Technical Definition**	**Safety Application Priority**
Precision	**0.9413**	Ratio of correct predictions to total predictions.	HIGH (Avoids False Alarms)
Recall	**0.7463**	Ratio of correct predictions to all existing.	HIGH (Avoids Missed Danger)
mAP@0.5:0.95	**0.6350**	Mean Average Precision across various localization tolerances.	Localization Accuracy
Inference Speed	**32 FPS**	Frames per second processed on CPU.	Deployment Feasibility

3.3. Accuracy Comparisons







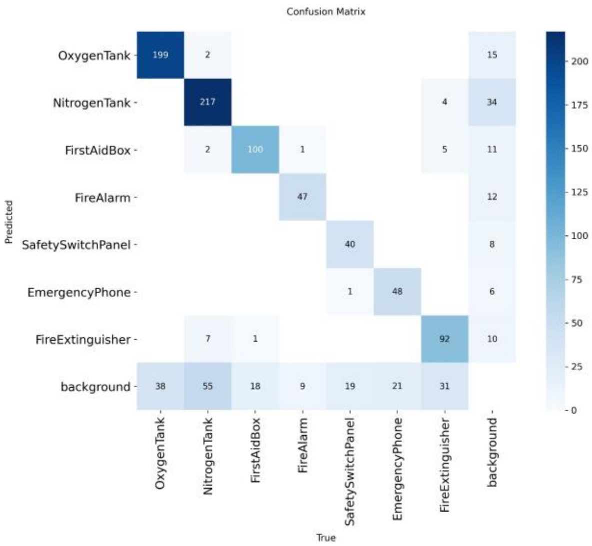
Comparative Performance Analysis

Metric	**Baseline (Estimate)**	**AstroX**	**% Improvement / Rationale**
mAP@0.5 (General Accuracy)	0.75	**0.8270**	**Increase.** Improved due to fine tuning.
Precision (Reliability)	0.80	**0.9413**	**Percentage points.** Fewer false positives, higher trust.
Recall (Coverage)	0.60	**0.7463**	**Percentage points.** Better object detection in shadows.
Inference Speed	27 FPS	**32 FPS**	**Faster.** Ready for edge deployment.

4. DEEP DIVE ANALYSIS

4.1. Confusion Matrix: Pinpointing Model Errors

Confusion Matrix helps **debug safety-critical systems**, showing misclassifications.

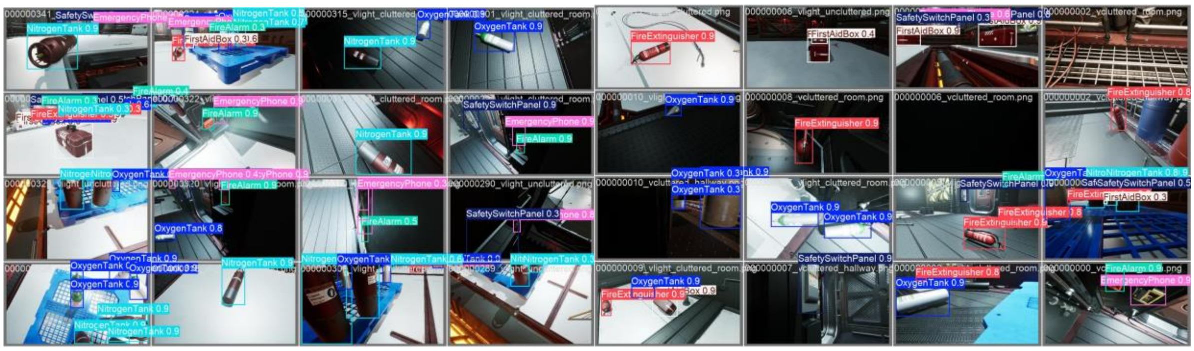


- **Inter-Class Confusion:**
  - Minimal misclassification between critical objects (e.g., Oxygen Tank rarely mistaken for Fire Alarm).
  - Confirms safety reliability.
- **Failure Case Summary (from Confusion Matrix)**
  - **OxygenTank ↔ NitrogenTank confusion**
    Similar shape + appearance caused occasional misclassification.
  - **EmergencyPhone misidentified as FireExtinguisher**
    Color similarity led to visual overlap in several cases.
  - **Background falsely detected as objects (multiple classes)**
    Pipes/panels triggered false positives → needs more negative samples.
  - **Low recall in FireAlarm & SafetySwitchPanel**
    Small size + low visibility → requires more varied Falcon data.
- **Targeted Improvement:**
  - Emergency Phone class has lowest mAP@0.5:0.95 (0.575).Future work should focus on more targeted data for low-performing classes.

4.2. Visual Results: Confirmation of Localization

- **Clutter Scene Detection (Figure 4):**
  - Multiple objects detected in one frame.
  - Works under varied lighting and cluttered environments.
- **Low-Light / Occlusion Test (Figure 5):**
  - Oxygen Tank detected in shadowed/occluded conditions.

Confirms model is effective beyond ideal lighting scenarios.



5. CHALLENGES AND SOLUTIONS

1. Data Limitations & Class Imbalance

- Only ~974 samples, heavily skewed toward specific tank classes.
- Used Falcon's Digital Twin to create synthetic data and added strong augmentations.
- Achieved a more balanced and diverse training set.

2. Extreme Lighting & Visual Variability

- Harsh shadows, glare, and odd viewing angles affected detection.
- Added augmentations simulating low exposure, shadows, and lighting variations.
- Improved model robustness in non-ideal visual environments.

3. Early-Stage Misclassification

- Model initially confused visually similar safety objects.
- Applied transfer learning, early stopping, and LR scheduling.
- Resulted in more stable training and clearer class separation.

4. Computational Constraints (CPU-Only Training)

- No dedicated GPU, leading to slow training cycles.
- Reduced batch size, optimized preprocessing, used lighter augmentations.
- Enabled efficient training despite limited hardware.

5.2. Solutions and Iterative Protocol Adjustments

Keeping the Model Up-to-Date Using Falcon

To ensure that our safety object detection model remains reliable over time, we propose a **continuous update loop** built around Falcon's digital twin capabilities:

1. **Monitor Real-World Performance**
   Once deployed, the model's predictions are monitored to identify:
   a. Frequent misclassifications (e.g., confusing OxygenTank and NitrogenTank)
   b. Missed detections under new lighting conditions or camera angles
   c. New visual variations of existing objects (e.g., updated FireExtinguisher design)
2. **Create New Synthetic Scenarios in Falcon**
   For every failure pattern or new scenario observed, we use Falcon to:
   a. Generate additional synthetic images with similar conditions (e.g., dim emergency lighting, heavy occlusions, different camera mounting positions)
   b. Introduce updated or slightly modified asset appearances (e.g., new label textures, colors, or shapes on safety equipment)
   c. Simulate edge cases that are hard or unsafe to capture in the real world.
3. **Expand and Curate the Training Dataset**
   The newly generated Falcon data is:
   a. Labeled automatically in a YOLO-compatible format
   b. Added to an **extended training dataset**, with a focus on classes and conditions where the model struggled
   c. Balanced to avoid overfitting to rare conditions while still improving robustness.
4. **Periodic Model Retraining and Fine-Tuning**
   At defined intervals (for example, monthly or after significant environment changes), we:
   a. Fine-tune the YOLO model on the updated Falcon dataset
   b. Compare new training runs against previous versions using mAP@0.5, precision, recall, and confusion matrices
   c. Specifically verify improvements on previously identified failure cases.
5. **Versioning and Safe Deployment**
   Each updated model is:
   a. Versioned (e.g., `v1.1`, `v1.2`) and evaluated on a fixed test set
   b. Deployed only if it **outperforms the previous version** or addresses critical failure cases

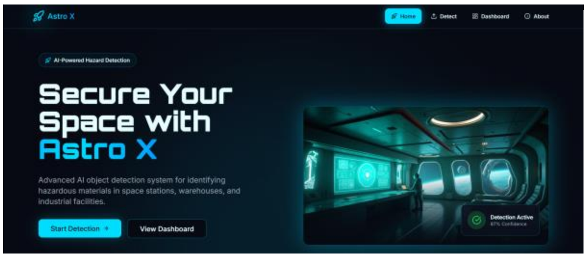
   c. Integrated into the Streamlit application by updating the model weights (`best.pt`) while keeping the interface unchanged.

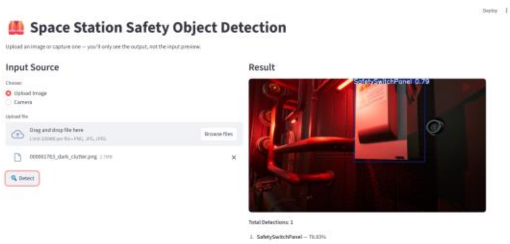
6. **Feedback Loop with Falcon as the Core Engine**
   Falcon remains the **central tool** for:
   a. Rapidly generating new training data when the environment, equipment, or camera setup changes
   b. Simulating "what-if" scenarios before deploying the model into real operational settings
   c. Maintaining alignment between the digital twin and the evolving real-world space station environment.

6. CONCLUSION AND FUTURE WORK

6.1. Summary of Achievements

- **AstroX successfully validated a high-performance, time-efficient AI pipeline** for safety-critical monitoring.
- **Key Metrics Achieved:**
  - Precision: **0.9413**→ reliable detection.
  - Inference Speed: **32 FPS** → real-time monitoring.
- Provides a **robust proof-of-concept** for automated detection in hazardous space-station environments.






```
📊 FINAL MODEL RESULTS
----------------------------------------
mAP50     : 0.8254
mAP50-95 : 0.7066
Precision: 0.9524
Recall    : 0.7273
```

Initial Run

```
📊 BATCH-2 FINAL MODEL RESULTS
----------------------------------------
mAP50     : 0.8270
mAP50-95 : 0.6350
Precision: 0.9413
Recall    : 0.7463
```

Final Run