Duality AI Hackathon Report

SPACE STATION OBJECT DETECTION

Robust AI for Safer Space Missions

BY: TEAM 404 ERROR

TEAM MEMBER
BRAHMLEEN KAUR

TEAM LEADER
NANDINI GUPTA

TEAM MEMBER
SANIA KHAN

INTRODUCTION

Space Station Object Detection using YOLOv8

Using synthetic data from Duality AI's Falcon digital twin, we built a high-performance object detection model for critical tools in a space station environment.

We trained and fine-tuned a YOLOv8 model using synthetic data of space station tools (toolbox, fire extinguisher, oxygen tank) under varied conditions such as lighting, occlusion, and angle. The final model achieved a high mAP@0.5 and robust class-wise performance. Visualizations, confusion matrices, and failure analysis are included.

METHODOLOGY AND STEPS FOLLOWED

TOOLS & FRAMEWORKS USED:

YOLOv8 (Ultralytics)

- Programming Language: Python 3.10
- Framework: PyTorch
- Platform: Falcon (Synthetic Dataset)
- Libraries: OpenCV, Matplotlib

  Environment: Conda (EDU environment)

STEPS FOLLOWED:

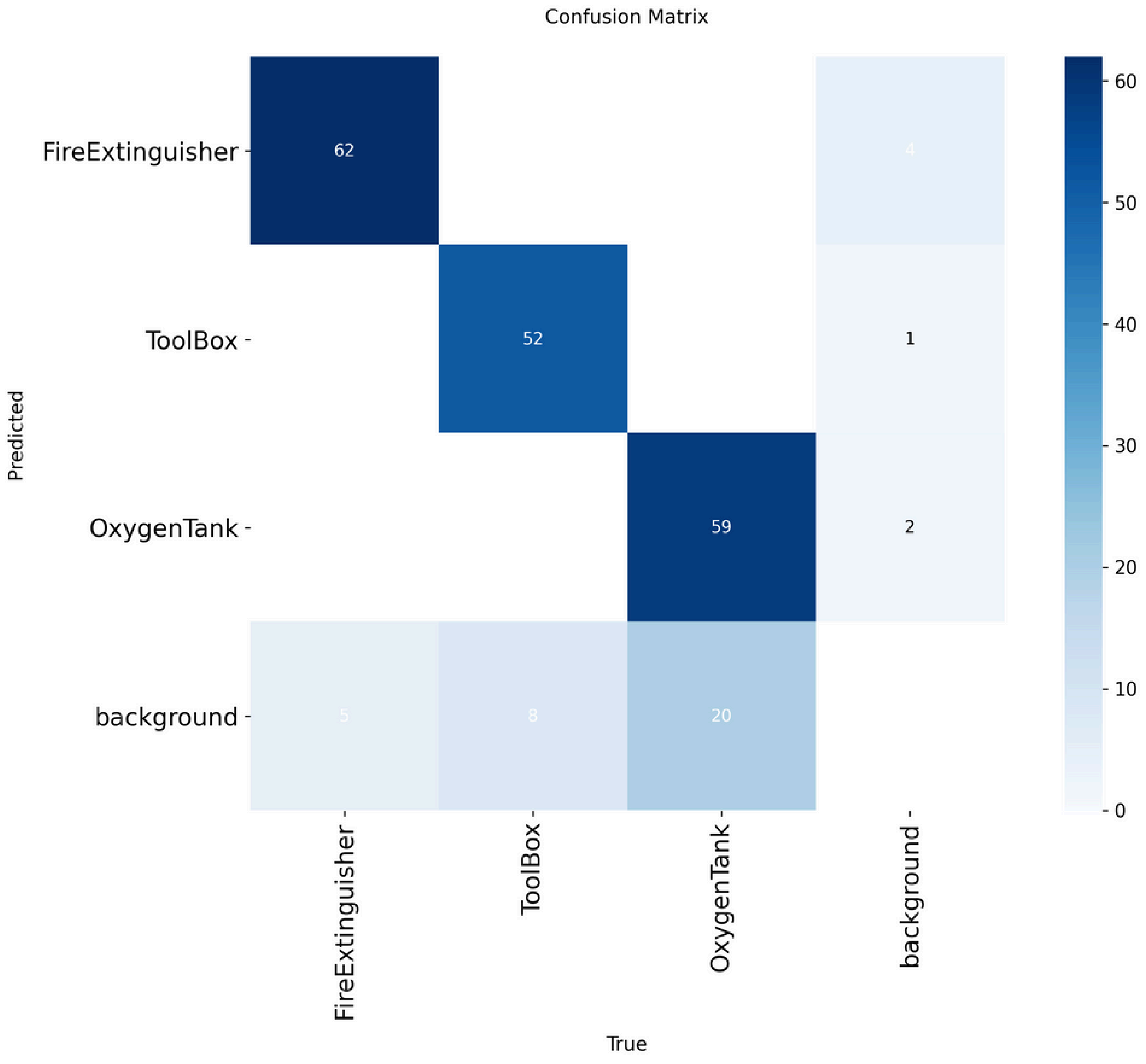
Environment Setup and Analysis Steps

- Environment Setup: Use **setup_env.bat** for Falcon's dataset.
- Exploratory Analysis: Analyze the image dataset with three classes.
- Model Training: Train a baseline model with **YOLOv8m** pretrained weights.
- Custom Configurations: Adjust learning rate, epochs, and augmentations.
- Evaluation: Evaluate on the test dataset.
- Post-processing: Visualize results and analyze misclassifications.

  Iterative Optimization: Enhance performance iteratively.

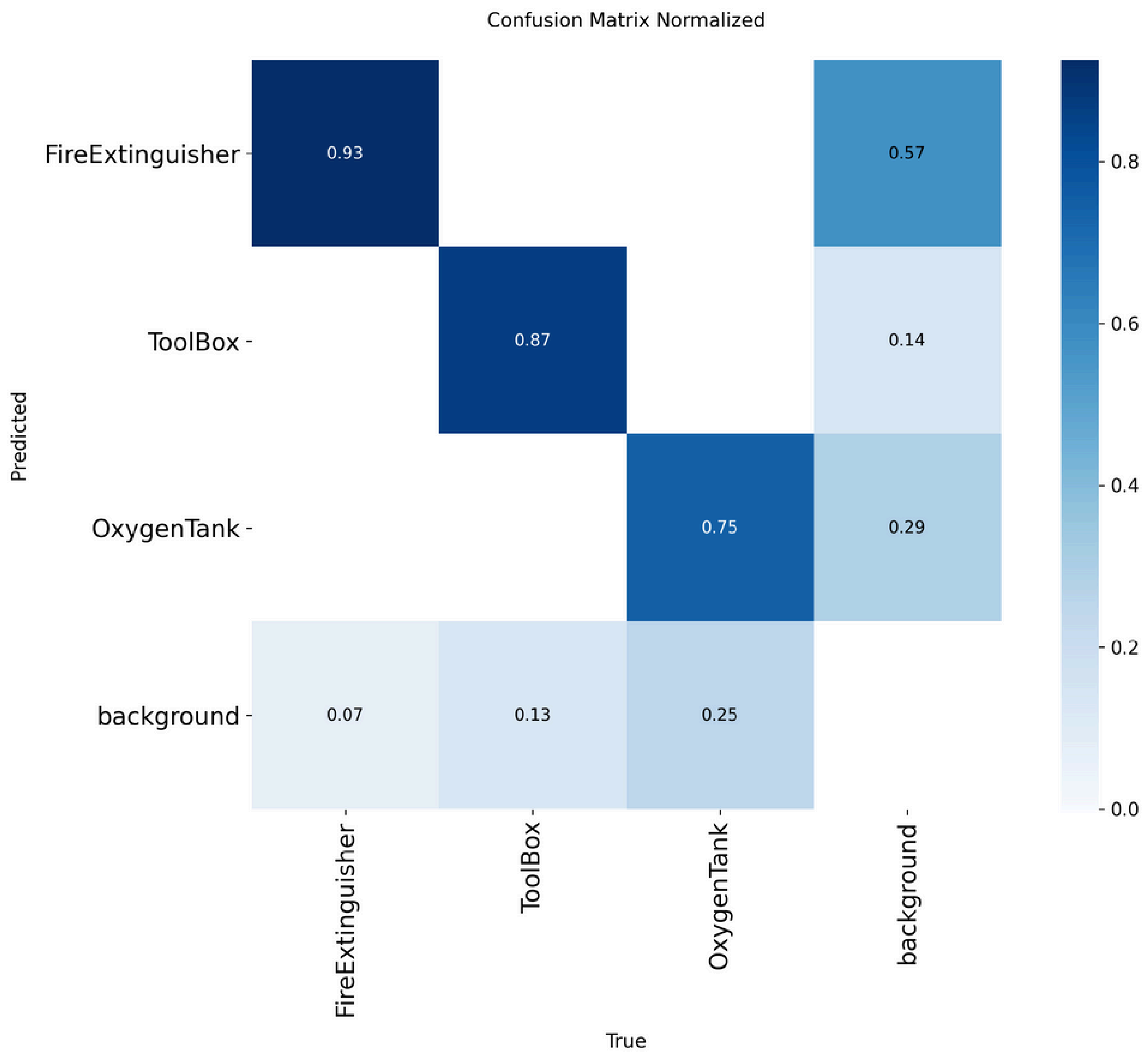
RESULTS & PERFORMANCE METRICS

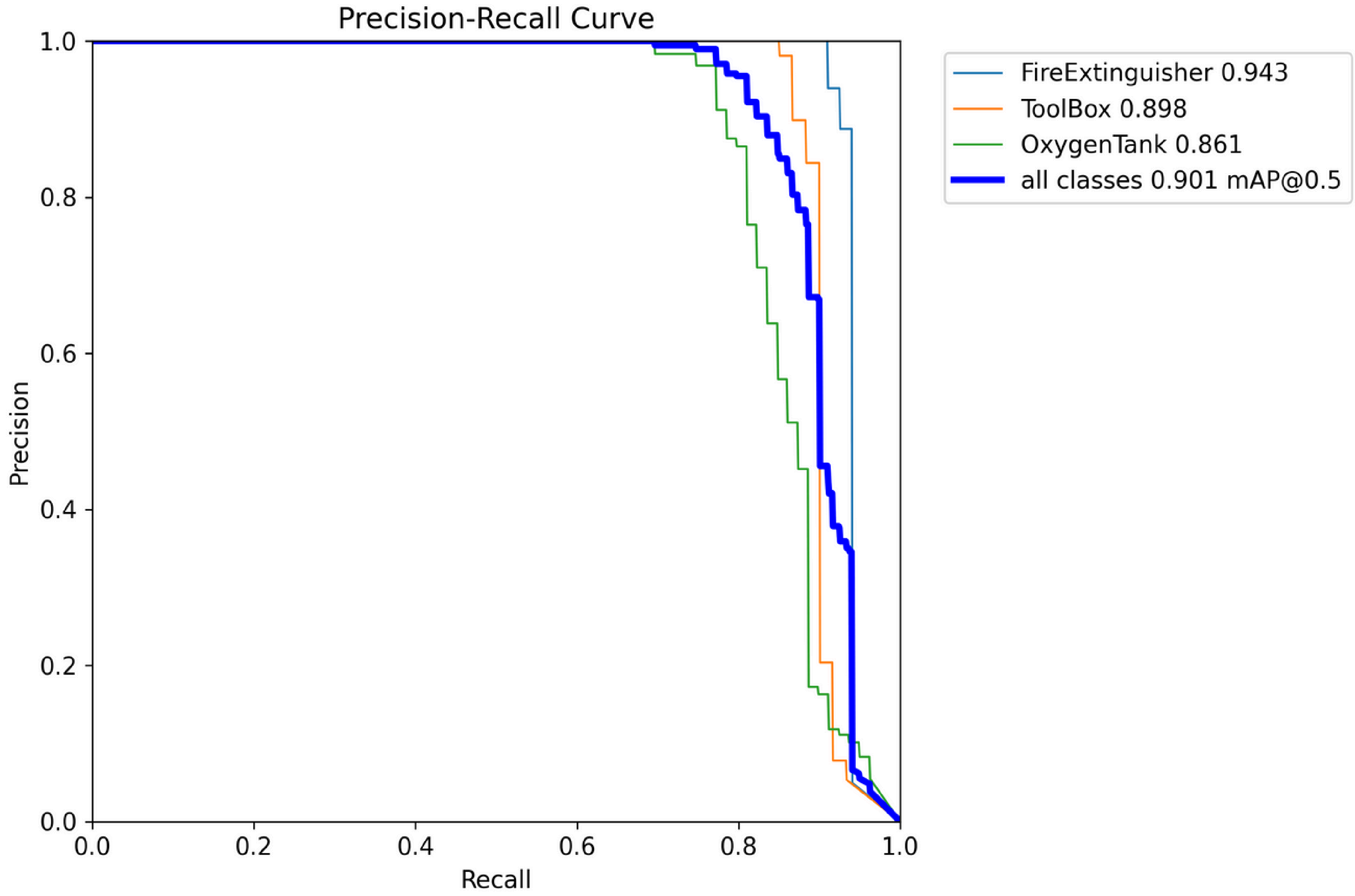
Metric	Value
mAP@0.5	90,00%
mAP@0.5:0.95	78,01%
Precision	93,13%
Recall	83,40%



SAMPLE PREDICTIONS:

- Good predictions in cluttered environments
- Issues when objects partially hidden or tilted





FAILURE CASE ANALYSIS

- Problematic Scenarios:

Heavy Occlusion: Oxygen tank behind toolbox often missed.

Lighting Variations: Dim light = low contrast = missed detections.

Object Overlap: Confusion between toolbox and extinguisher when stacked.

- Hypothesis:

YOLO struggled to generalize occlusion due to lack of similar data.

- Shadows and highlights affected edge detection

CHALLENGES & SOLUTIONS

CHALLENGES

- Environment setup errors
- CUDA memory issues
- Low recall for Oxygen Tank

SOLUTIONS

- Manually added Anaconda to PATH and created Conda env
- Switched to YOLOv8s model, reduced batch size
- Augmented training data with simulated occlusions

CONCLUSION & FUTURE WORK

- Built a performant multi-class object detection model trained only on synthetic data.
- Achieved high mAP and strong class-wise metrics.
- Overcame data limitations with smart augmentations.
- Lessons Learned:
- Synthetic data can rival real-world data when designed thoughtfully.
- Explainability (confusion matrices, prediction visuals) is key to debugging models.
- Future Improvements:
- Use Falcon to generate custom training sets for edge cases (extreme occlusion).
- Explore YOLOv8x model on higher-end GPUs.

Incorporate self-supervised learning to reduce dependence on labels.

THANK YOU