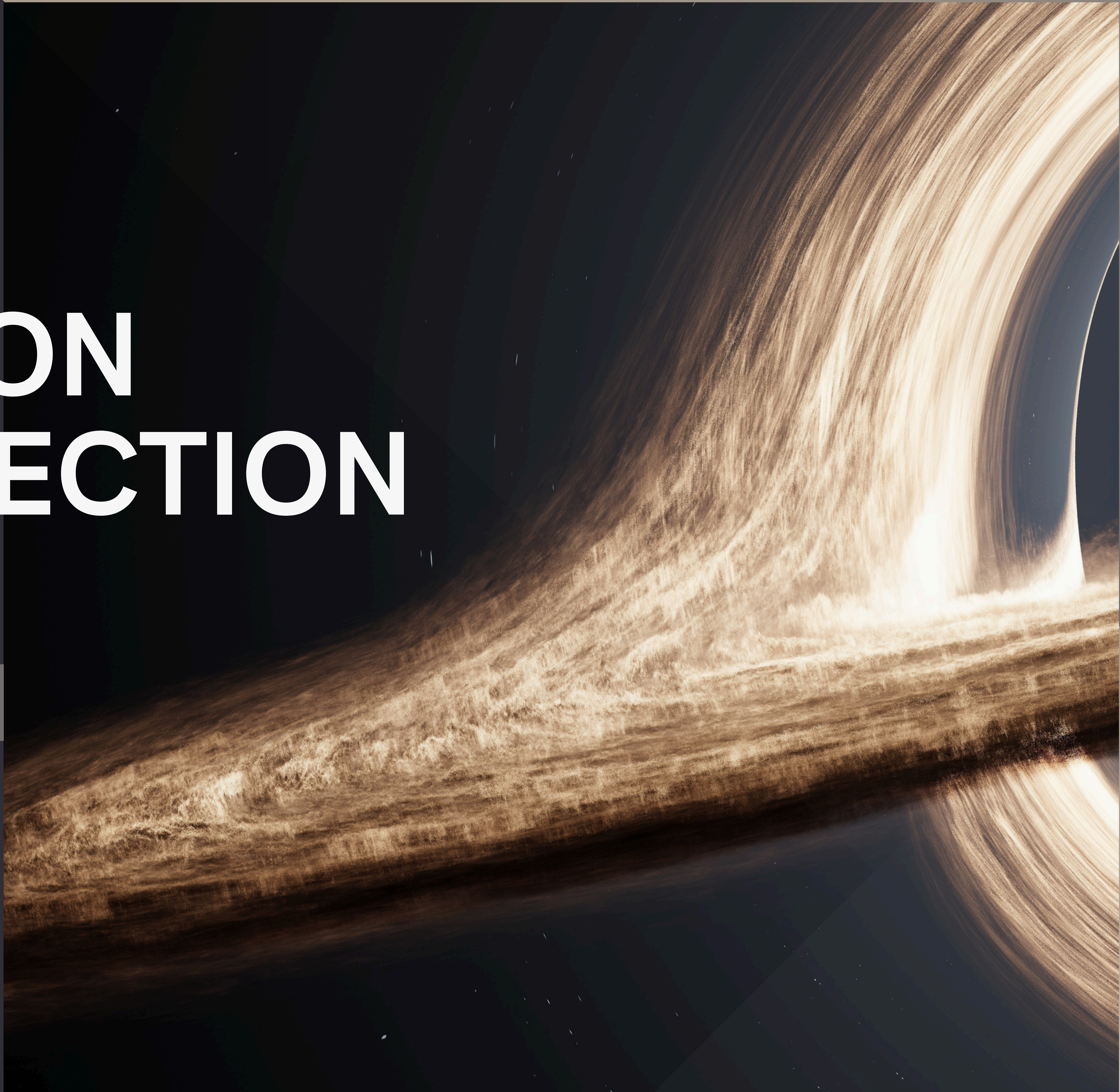
An Introduction to Object Detection in Space

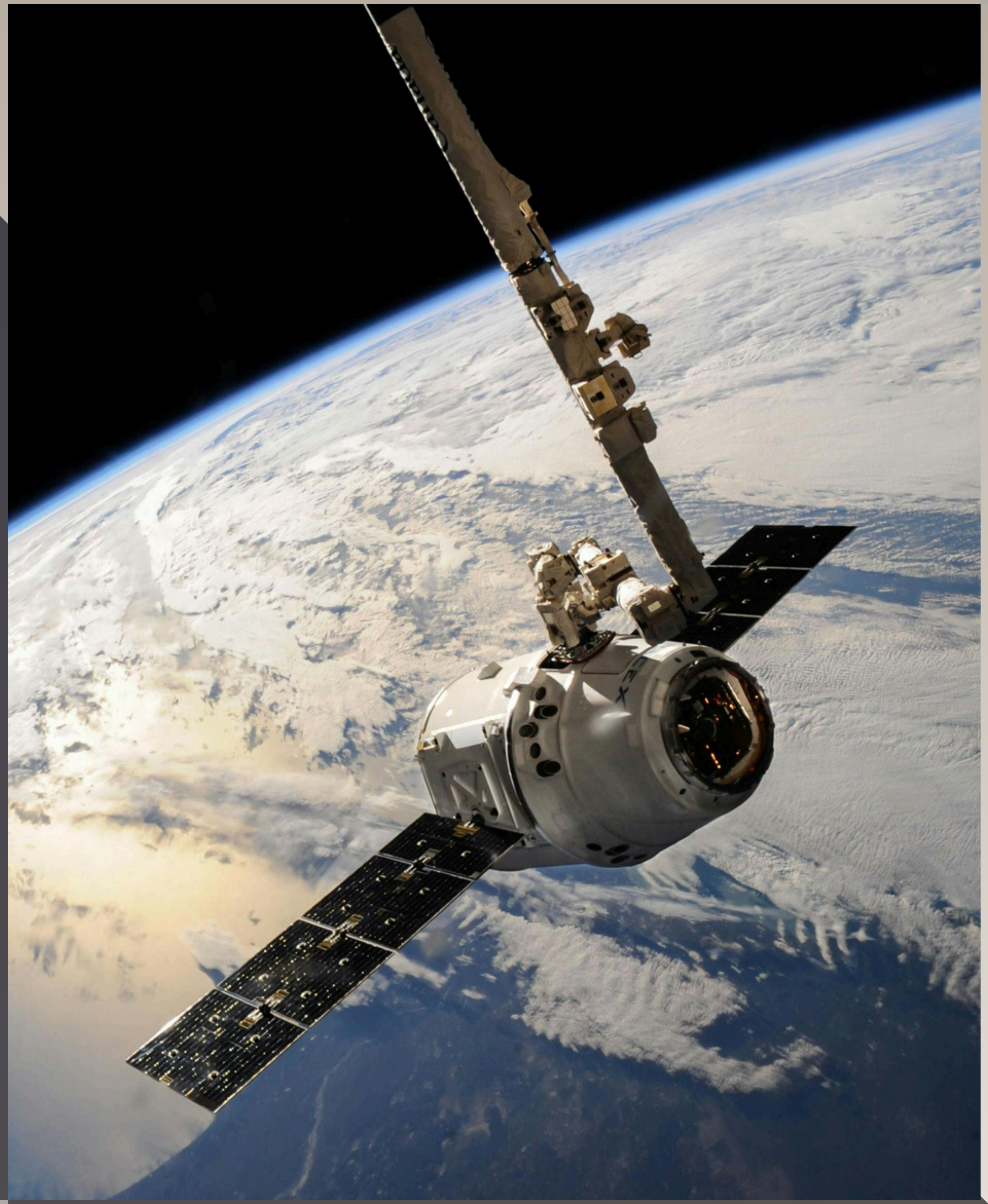
SPACE STATION OBJECT DETECTION



Team Name: HOPE

TEAM MEMBERS:

Damanjeet Singh
Achint Singh
Dilpreet Singh



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: Techniques and Processes for Detection

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.

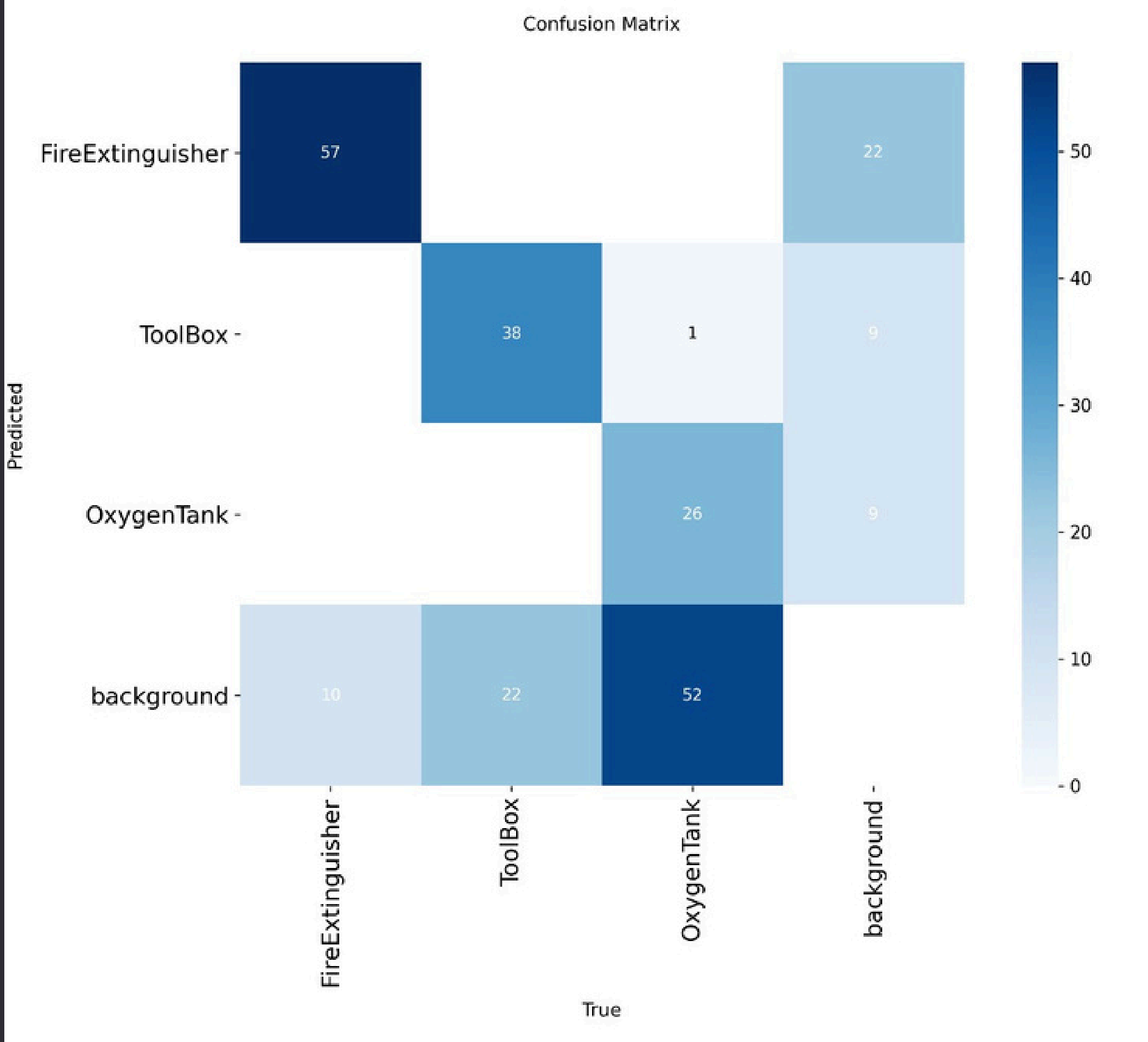
Iterative Optimization: Enhance performance iteratively.

RESULTS AND PERFORMANCE METRICS

MAP@0.5

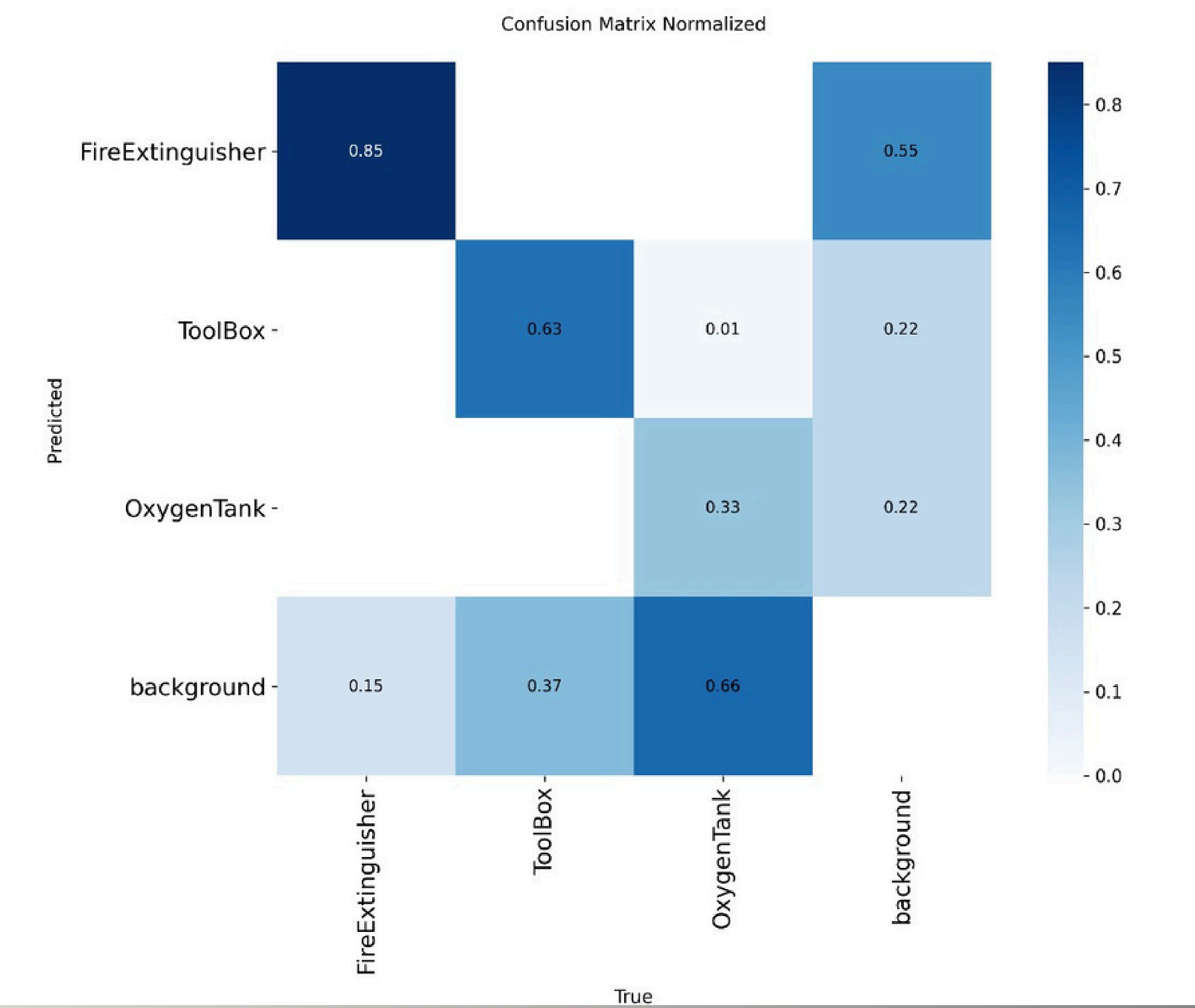
Metric	Value
mAP@0.5	5.22%
mAP@0.5:0.95	2.00%
Precision	9.17%
Recall	14.00%

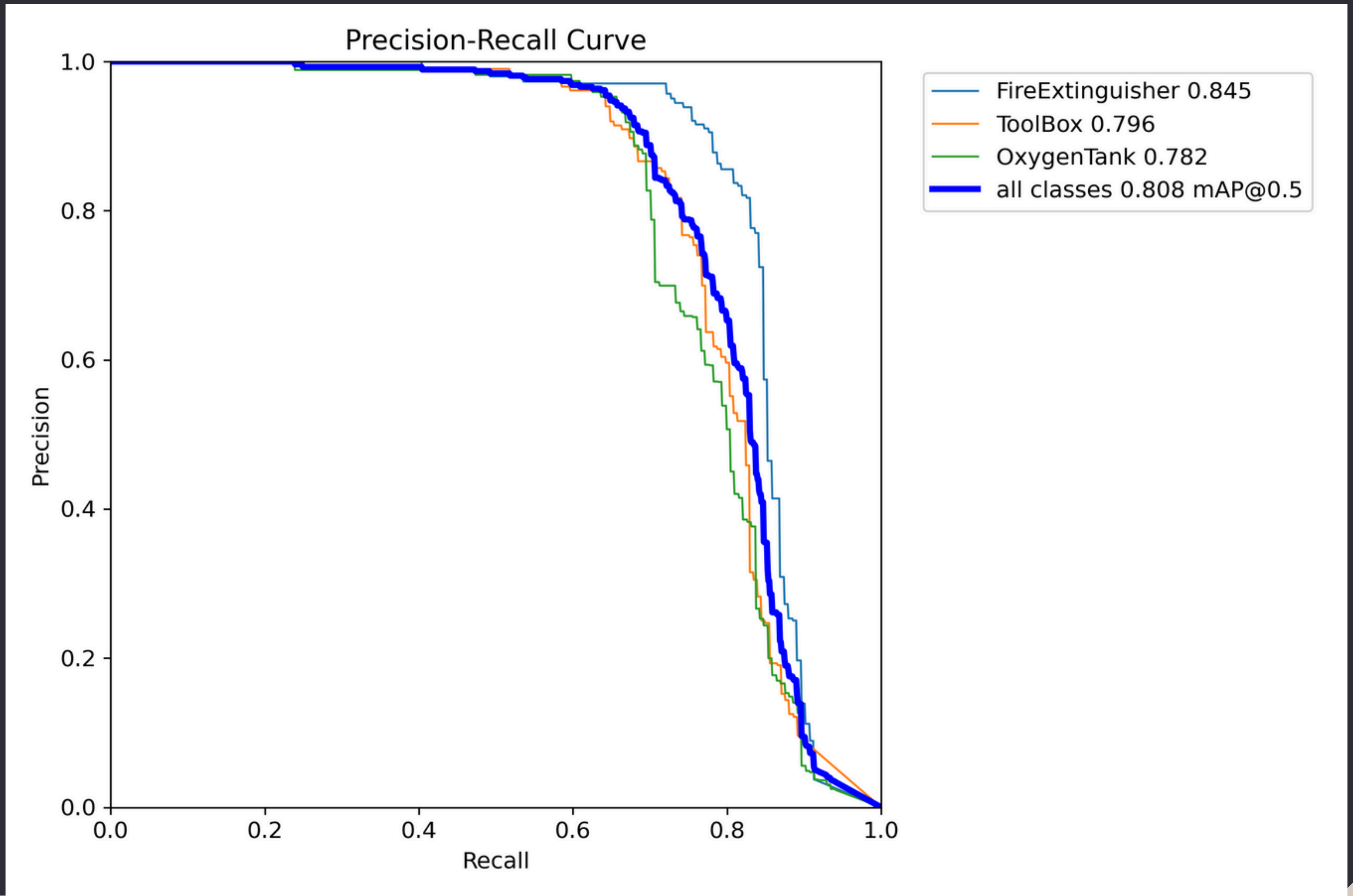
CONFUSION MATRIX



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 AND 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 AND FUTURE WORK

- Built a performant multi-class object detection model trained only on synthetic data.

1. Achieved high mAP and strong class-wise metrics.
2. Overcame data limitations with smart augmentations.

- Lessons Learned:

1. Synthetic data can rival real-world data when designed thoughtfully.
2. Explainability (confusion matrices, prediction visuals) is key to debugging models.

- Future Improvements:

1. Use Falcon to generate custom training sets for edge cases (extreme occlusion).
2. Explore YOLOv8x model on higher-end GPUs.

Incorporate self-supervised learning to reduce dependence on labels.