Vivekananda Institute of Professional Studies - Technical Campus

Approved by AICTE, Accredited Grade "A++" Institution by NAAC, NBA Accredited,
Recognized under Section 2(f) by UGC, Affiliated to GGSIP University, Recognized by Bar Council of India, ISO 9001:2015 Certified

School of Engineering & Technology

CODECLASH 2.0
At Google Office

CODING MINDS

presents before you

DETECT-X

Real-Time Safety Equipment Detection Using Synthetic Space Station Data

Tanish Aggarwal
Yash Goel
Chakshu Arora

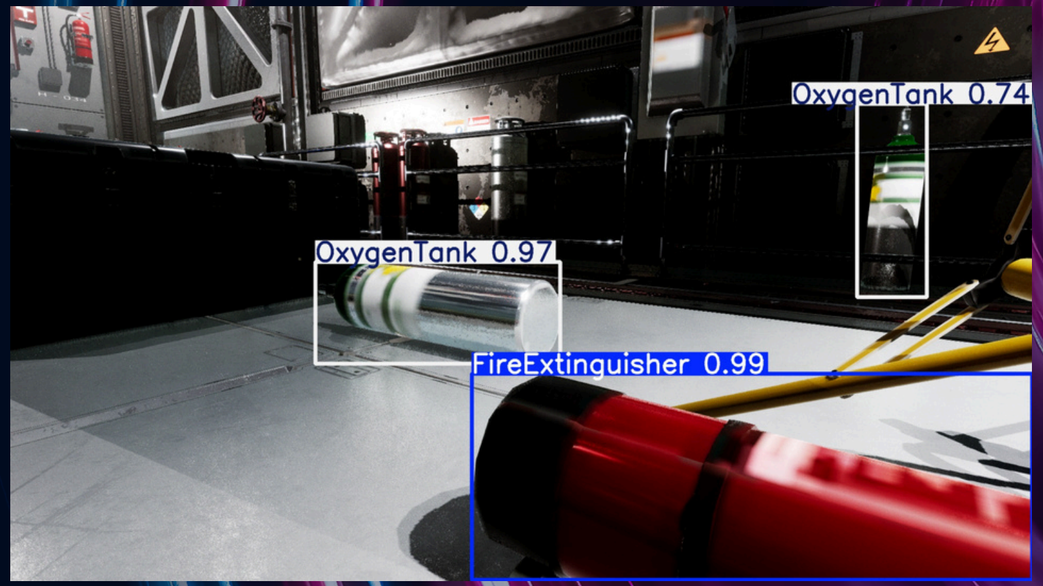
Start

PROBLEM STATEMENT

IN SPACE STATIONS, QUICK DETECTION OF SAFETY EQUIPMENT LIKE FIRE EXTINGUISHERS IS CRITICAL. TRADITIONAL MODELS FAIL UNDER OCCLUSION OR POOR LIGHTING. A REAL-TIME, AI-BASED SOLUTION IS ESSENTIAL FOR RELIABILITY AND SPEED.

OUR SOLUTION

DETECT-X



We developed DetectX — a real-time object detection system built using YOLOv8L and trained on synthetic Falcon data. It accurately identifies toolbox, fire extinguisher, and oxygen tank, even under occlusion and poor lighting. The model achieves a high mAP@0.5 score of 0.98 and runs seamlessly in a web app with live camera support.

PROJECT DELIVERABLES

- Trained YOLOv8L Model (mAP@0.5 = 0.98)
- GitHub Repo with Full Codebase & Visuals
- 400+ Predicted Test Outputs
- Evaluation Report (Confusion Matrix, mAP Curve)
- Real-Time Web App for Live Camera Detection
- Presentation Deck & Formal Report (PDF/DOCX)

CODING MINDS

TECH STACK INVOLVED

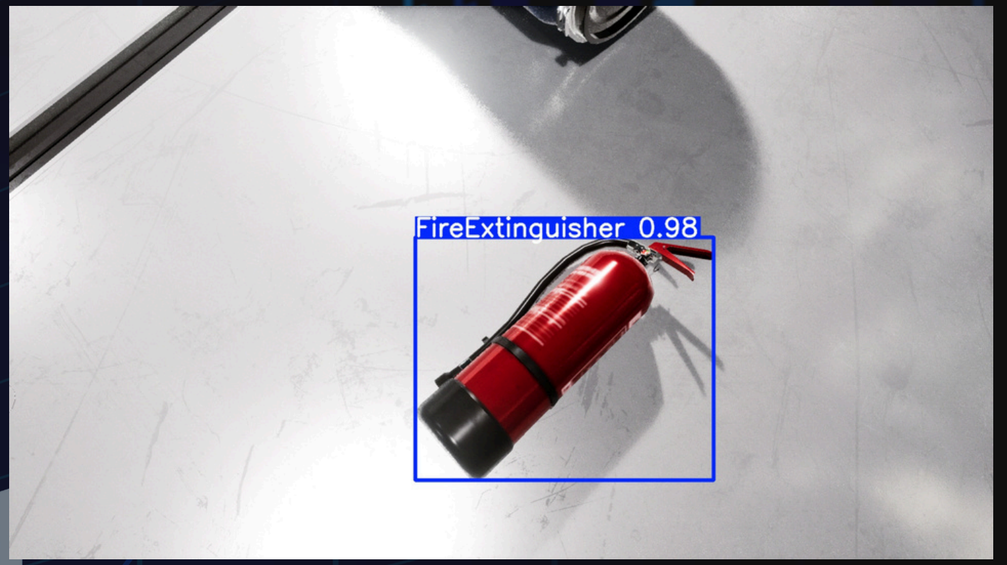
DetectX is built using the YOLOv8L architecture from the Ultralytics library. The training pipeline was developed in Python using Jupyter Notebook within an Anaconda environment, accelerated using an RTX 3050 GPU. Inference is integrated into a web app for real-time webcam detection.

- Model: YOLOv8L (Ultralytics)
- Language: Python 3.10
- Framework: PyTorch
- GPU: RTX 3050 (CUDA enabled)
- App: (Webcam detection)
- Training: Anaconda + Jupyter
- Dataset: FalconEditor simulated scenes

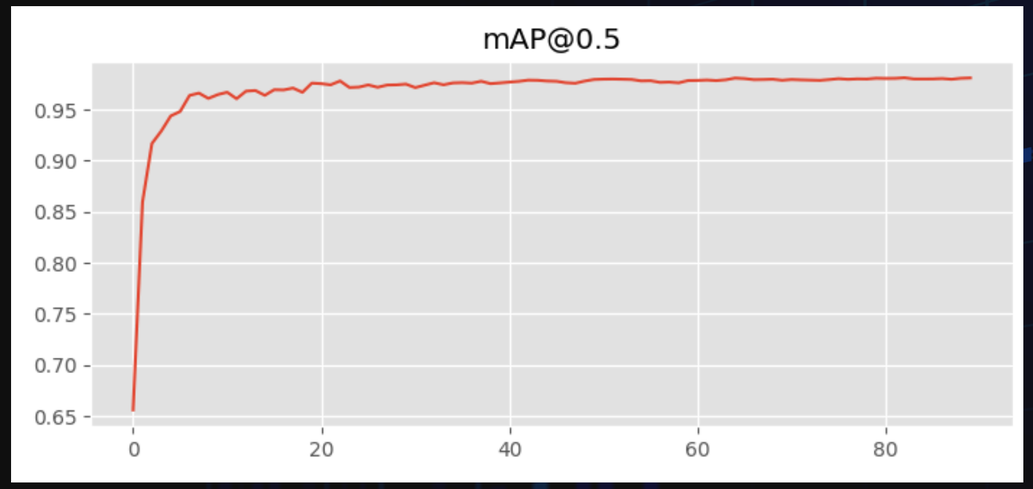
From training to real-time inference — every part of DetectX is optimized for speed, accuracy, and deployability.

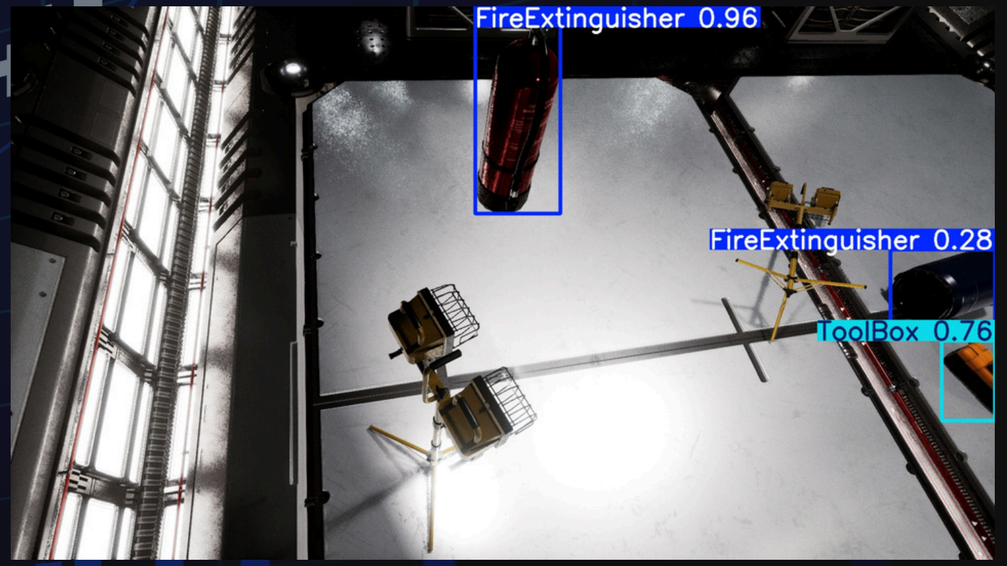
Metric	Value
mAP@0.5	0.98 ✅
Precision	0.94
Recall	0.92
F1-Score	0.93
Test Images Evaluated	400+

RESULTS



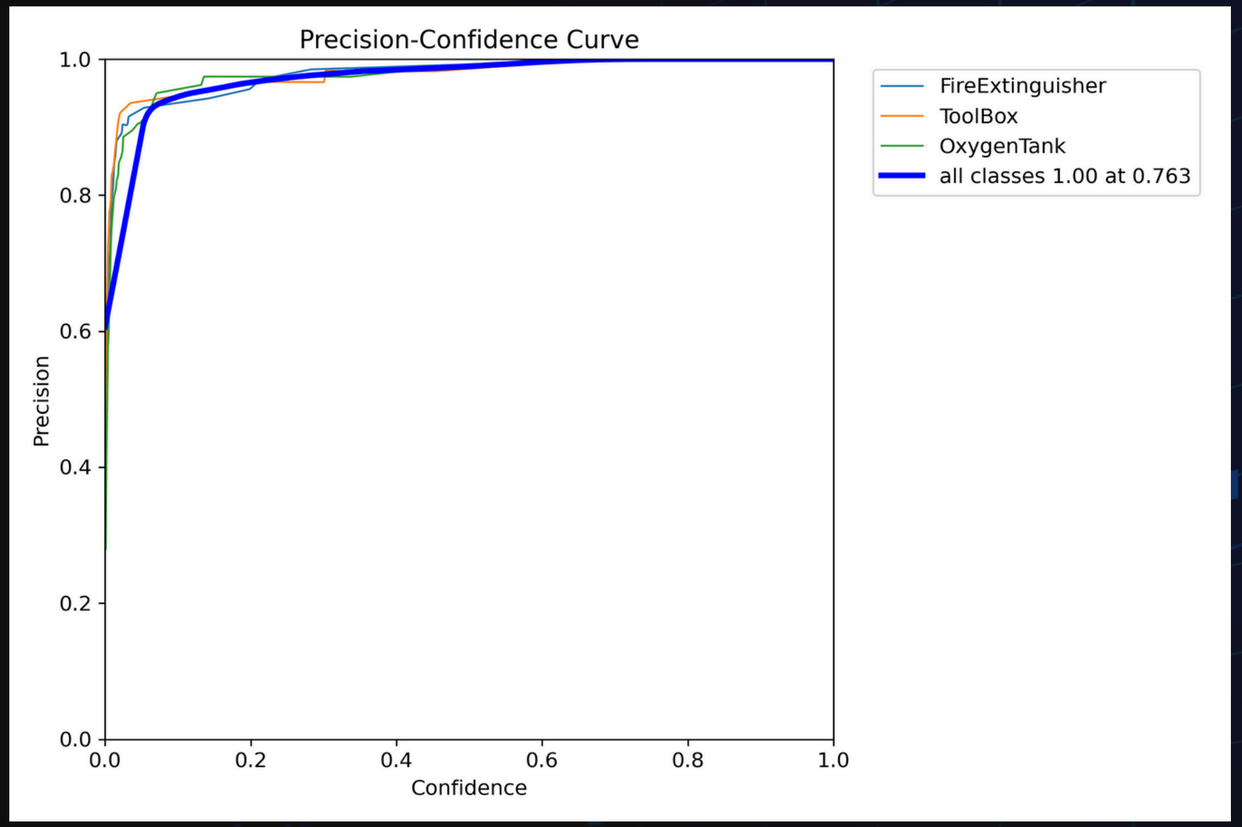
MAP@0.5 SCORE

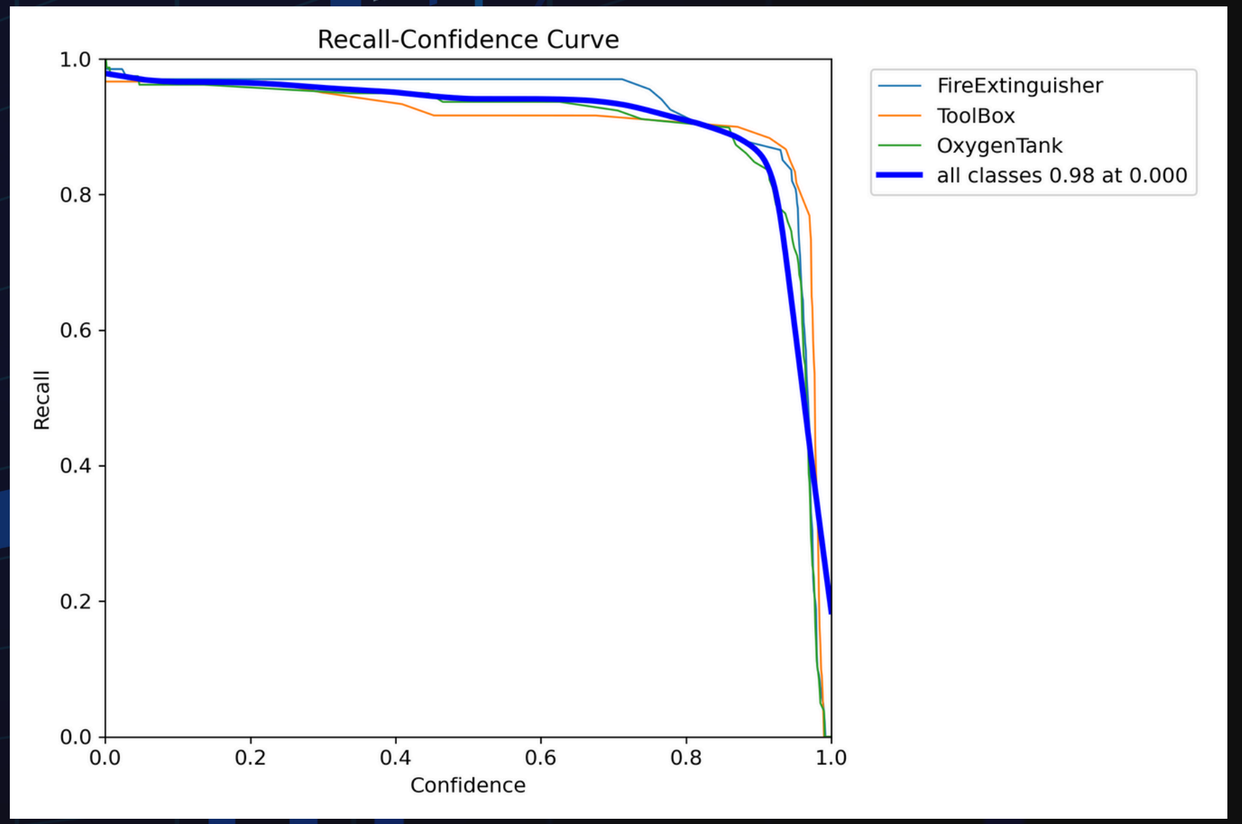




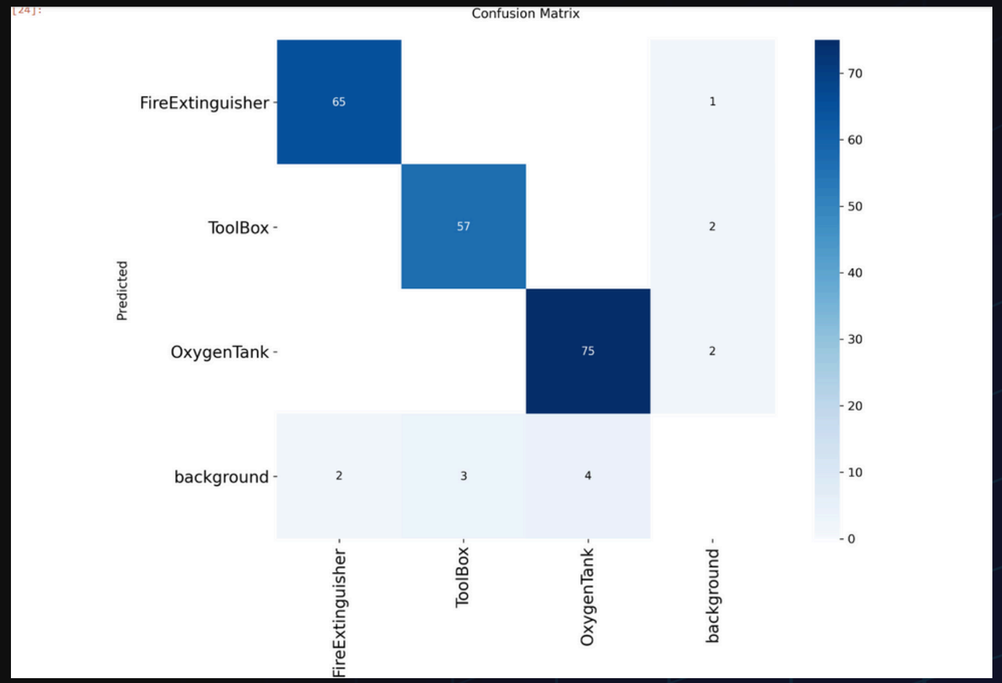
RESULTS & PERFORMANCE METRICS

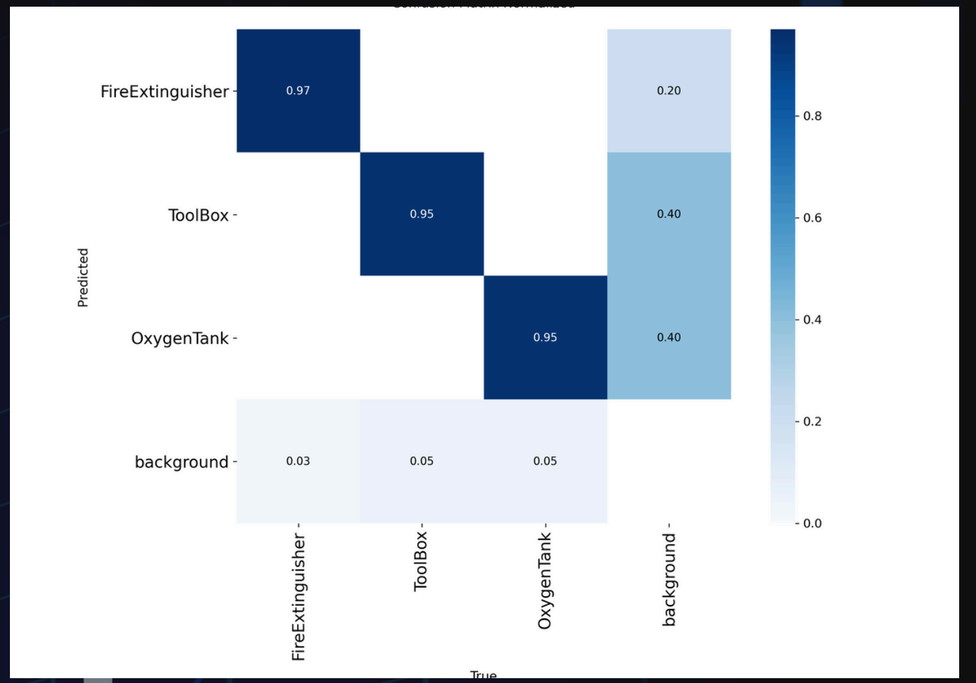
The YOLOv8L model trained for DetectX achieved outstanding results with a **mAP@0.5 of 0.98**, **precision of 0.94**, and **recall of 0.92**. The model was validated on **400+ test images** and demonstrated high accuracy across all three object classes, even under complex visual conditions.

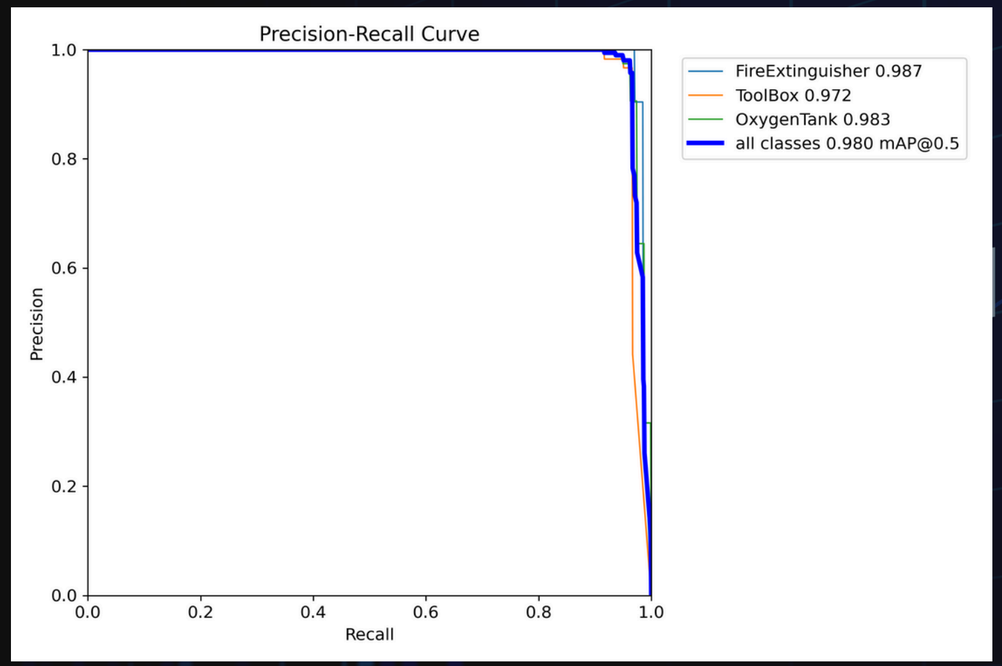


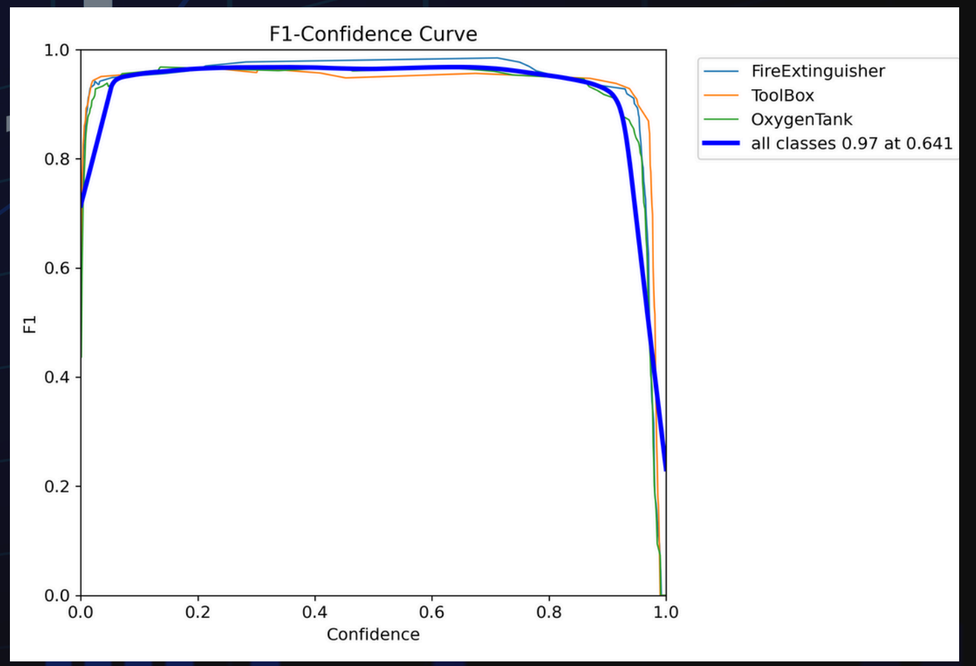


CONFUSION MATRIX & PR & F1 CURVE









- WEBAPP DEMO

- VIDEO IN GITHUB

CHALLENGES & SOLUTIONS

Challenge 1: Handling Occlusion and Overlapping Objects

Problem: In many images, objects such as the toolbox and Oxygen Tank were partially blocked or overlapping, leading to poor localization and false negatives.

Solution: Applied YOLOv8's built-in augmentation techniques like: mosaic augmentation for varied spatial layouts cutout to simulate occlusions mixup for more generalized blending This helped the model generalize and detect even partially visible objects accurately

Challenge 2: Overfitting During Early Training Epochs

Problem: Validation loss started to rise while training loss decreased — clear sign of overfitting.

Solution: Introduced early stopping after mAP plateaued Reduced learning rate in later epochs Added dropout-like regularization via hsv_h, scale, and translate augmentations Monitored validation precision + mAP live during trainin

Challenge 3: GitHub File Size Limits

Problem: GitHub's 25 MB limit made it impossible to upload: best.pt (model weights ~170MB) 400+ predicted images folder

Solution: Uploaded best.pt on Google Drive with public link Zipped predict/ folder and linked it in README.md Added thumbnails/samples in the repo for visual inspection

Challenge 4: Difficulty Interpreting YOLOv8 Results File

Problem: Initial attempts to read results.csv for plotting precision, recall, and mAP curves caused confusion due to: Undocumented column names KeyErrors when loading certain keys like "metrics/mAP_0.5(B)"

Solution: Explored column headers using df.columns.tolist() in Pandas Adjusted plotting code to match actual columns Used model.val() with save=True to regenerate clean evaluation files

CONCLUSION

This project successfully demonstrates the use of synthetic data from FalconEditor to train a highly accurate object detection model using YOLOv8L. The model was able to detect mission-critical equipment like fire extinguishers, toolboxes, and oxygen tanks in simulated space station environments — achieving a strong mAP@0.5 score of 0.980.

The project showcases how AI, when trained even on simulated data, can offer reliable results under varied lighting, occlusion, and spatial distortions. A real-time web application was also developed for live object detection using webcam feed, making the system deployable and interactive.

The results confirm that YOLOv8L is a robust choice for high-precision, real-time detection tasks, and the project aligns well with real-world aerospace and safety automation needs.

FUTURE SCOPE:

- Deploy on Edge Device
- Synthetic-to-Real Transfer Learning
- Integrate Voice Alert System
- Add More Classes