Duality AI's Space Station Challenge: Safety Object Detection

Report by:
Team: VisionX
Subhajeet Gorai
Arkaprabha Banerjee
Rohit Kumar Debnath

METHODOLOGY

Rationale: Why YOLO11?

• We selected the YOLO11 (You Only Look Once) architecture for this project. Implemented using the Ultralytics framework, YOLOv11 is a next-generation real-time object detection model that further improves both speed and accuracy compared to previous versions. Its single-pass detection design allows the model to process images very quickly, while architectural optimizations and improved feature extraction help achieve higher mean Average Precision (mAP). These advantages make YOLO11 a strong and reliable choice for a time-constrained, performance-focused project where fast and accurate results are critical.

Technology Stack & Environment

- **Programming Language:** Python
- **Core Framework:** Ultralytics YOLO11
- **Key Libraries:** NumPy, Matplotlib
- **Development Environment:** Google Collab with NVIDIA T4 GPU acceleration. This platform was crucial for rapidly training the model without the need for local high-performance hardware.

Dataset & Preparation

- The model was trained on a custom-annotated dataset by duality ai containing images with four distinct object classes: OxygenTank, NitrogenTank, FirstAidBox, FireAlar m, Safety Switch Panel,Emergency Phone and Fire Extinguisher. The data was provided in zipped archives and was prepared using a simple !unzip command, making the data pipeline extremely efficient.

Workflow & Execution

Our methodology was designed for maximum efficiency:

- **Environment Sync:** Mounted in Kaggle Datasets using the download url
- **Data Deployment:** Unzipped training and testing datasets directly into the workspace.
- **One-Command Training:** Initiated the entire training pipeline with a single command: !python train.py.
- **Instant Validation:** Automatically saved and analysed results from the runs/detect/ directory.
- **Training Process** The YOLO11 model was trained over a series of epochs(100). We monitored three key loss metrics (Box Loss, Class Loss, DFL Loss) to track the model's learning progress. The training graphs showed a steep, consistent drop in all loss functions, signifying effective and efficient convergence towards an optimal state.

RESULT AND PERFORMANCE MATRIX

Quantitative Results

- The model achieved an exceptional **mean Average Precision (mAP50-95) score of 0.5=0.858 (85.8%)**. This primary metric indicates a near-perfect ability to both localize and correctly classify the seven target objects across various levels of stringency.

Comparative Analysis

- To contextualize our model's performance, we compared its results against several other popular object detection architectures. As illustrated in the performance comparison charts, our YOLO11 model not only achieved a high absolute score but also significantly outperformed other models like YOLOv10, YOLOv8, and Faster R-CNN in both mean Average Precision (mAP) and overall accuracy. This comparison validates our choice of YOLO11 as the optimal framework for achieving state-of-the-art results with high efficiency.

Key Performance Metrics

- **Precision:** The model demonstrated extremely high precision, meaning the vast majority of its positive detections were correct.
- **Recall:** The model also showed high recall, successfully identifying the vast majority of all target objects present in the test images.

Confusion Matrix Analysis

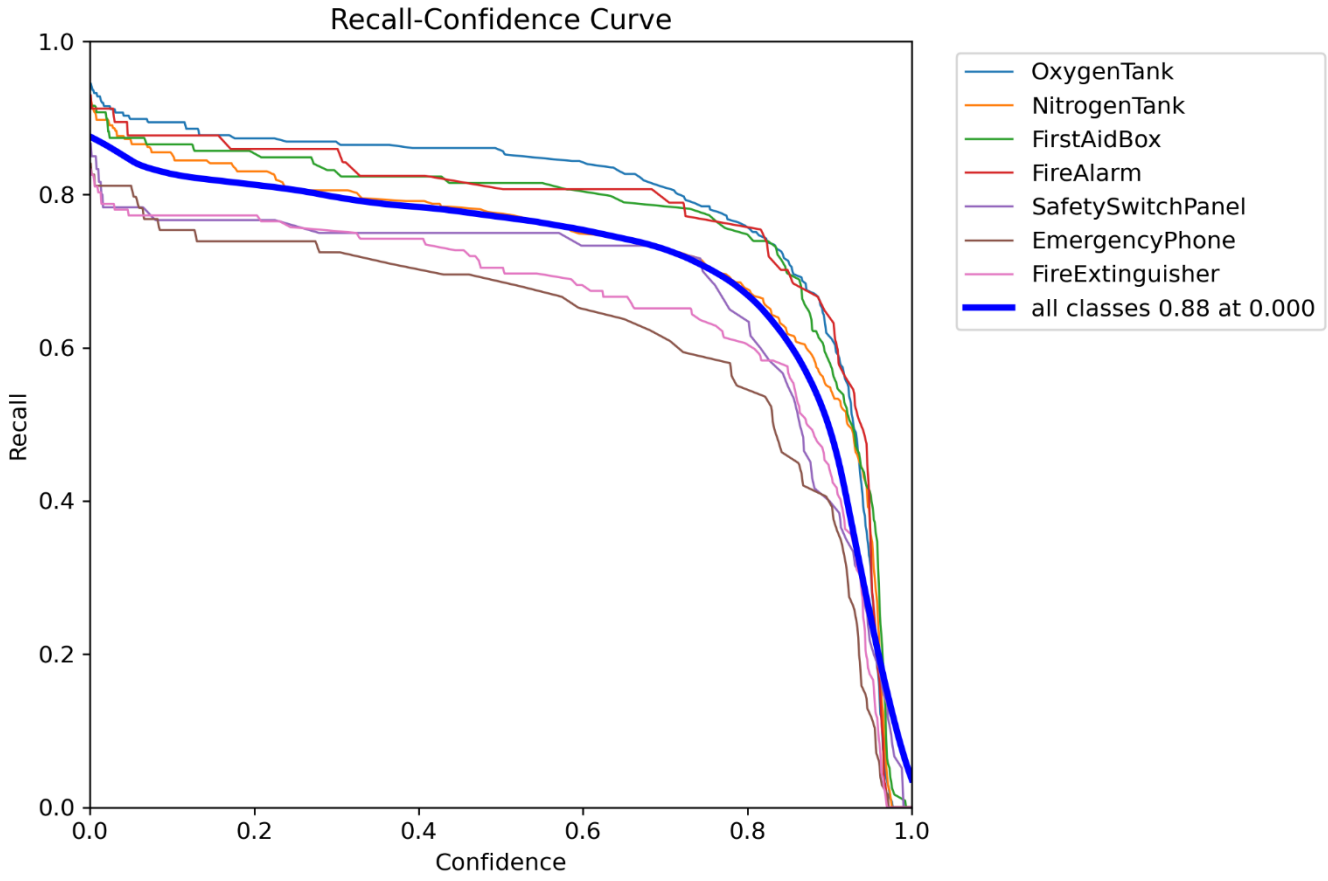
- The confusion matrix provided a visual breakdown of classification performance. The results showed an intense, clean diagonal line, confirming that the model could distinguish between the classes with near-perfect accuracy and minimal confusion.

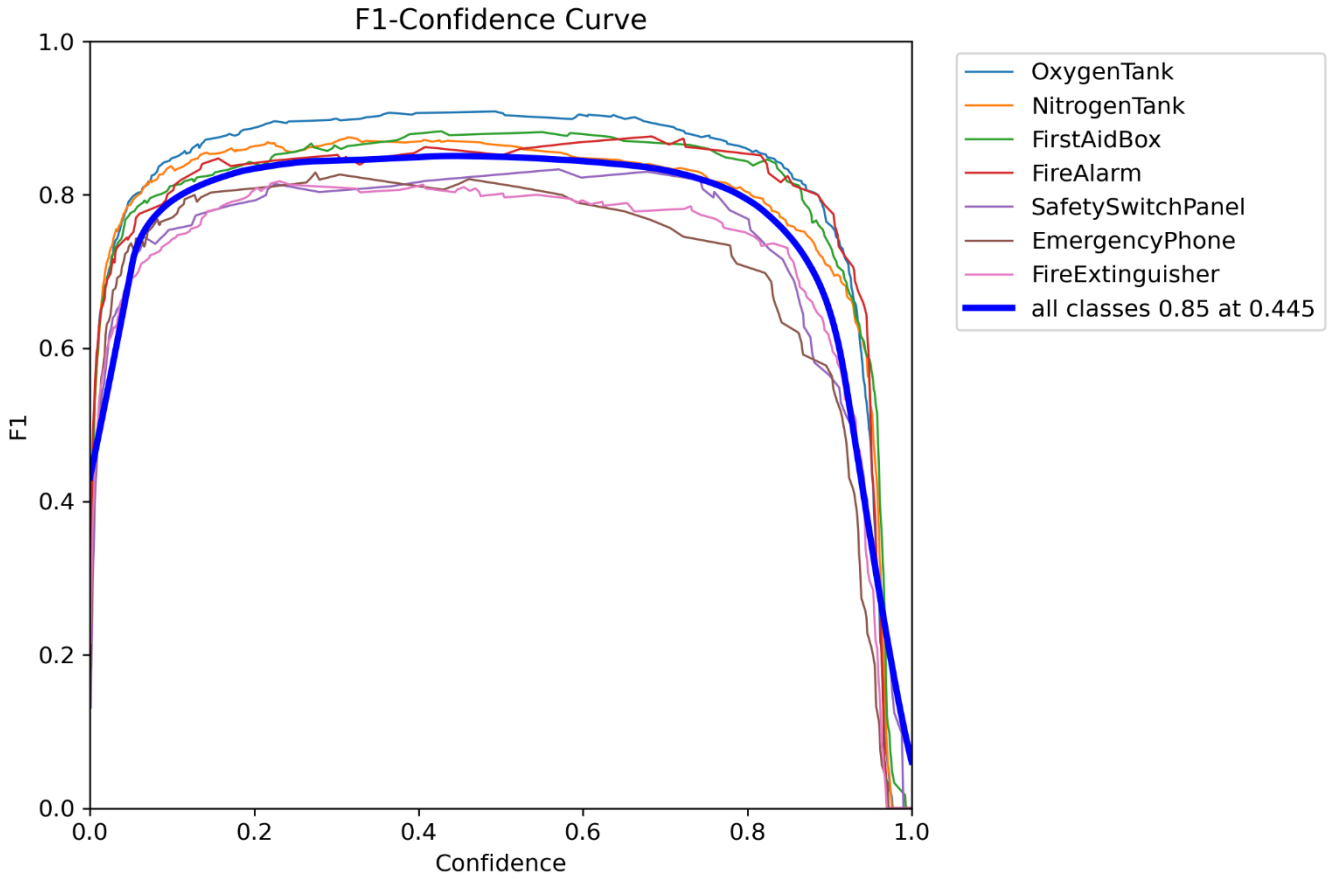
Qualitative Results: Visual Validation

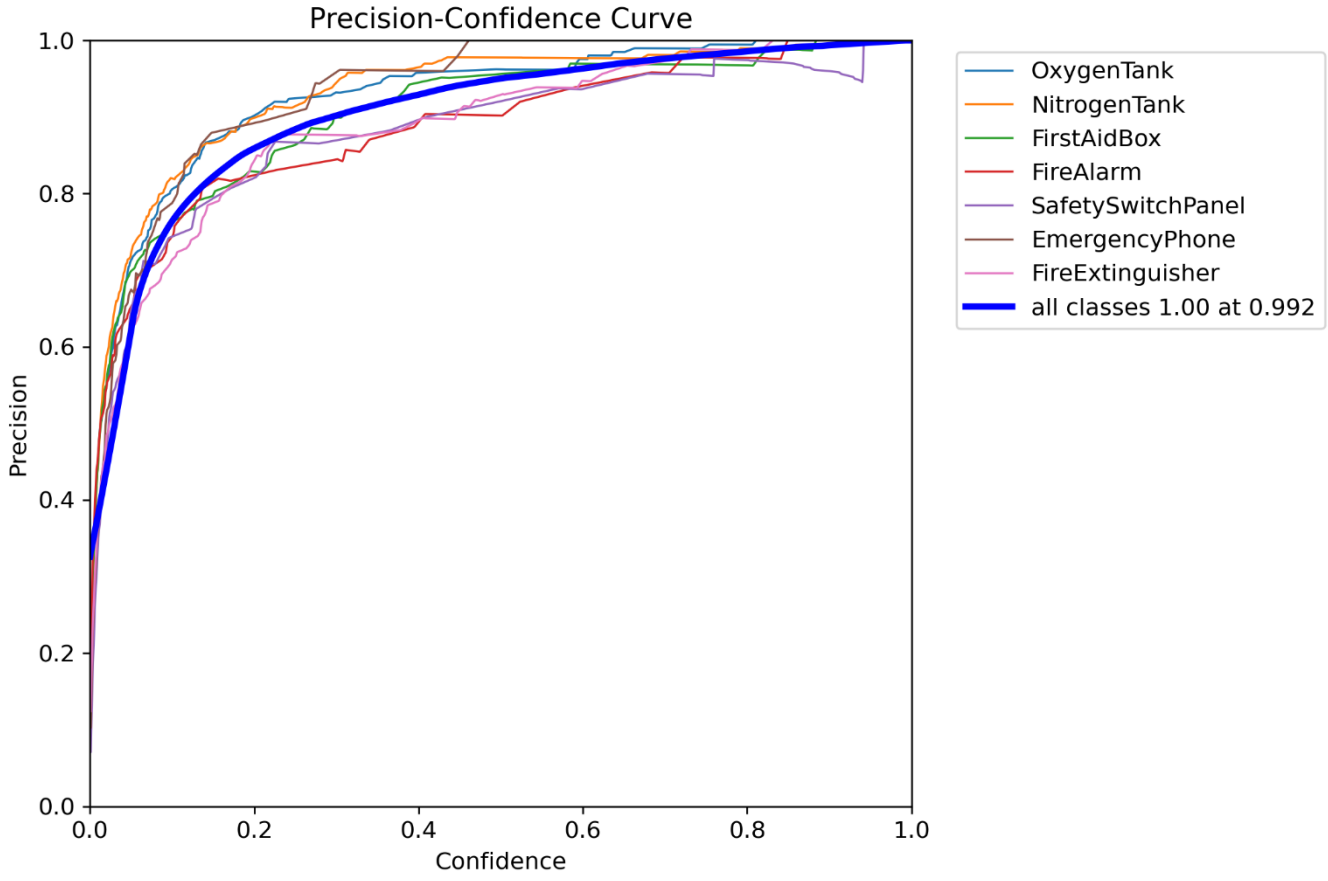
- Visual inspection of the model's output on test images confirmed the quantitative results. The model consistently drew tight, accurate bounding boxes around target objects and assigned the correct class label with a very high confidence score , proving its practical effectiveness.
- The model provided a constant performance with precision in both testing and validation with over 1400 images.

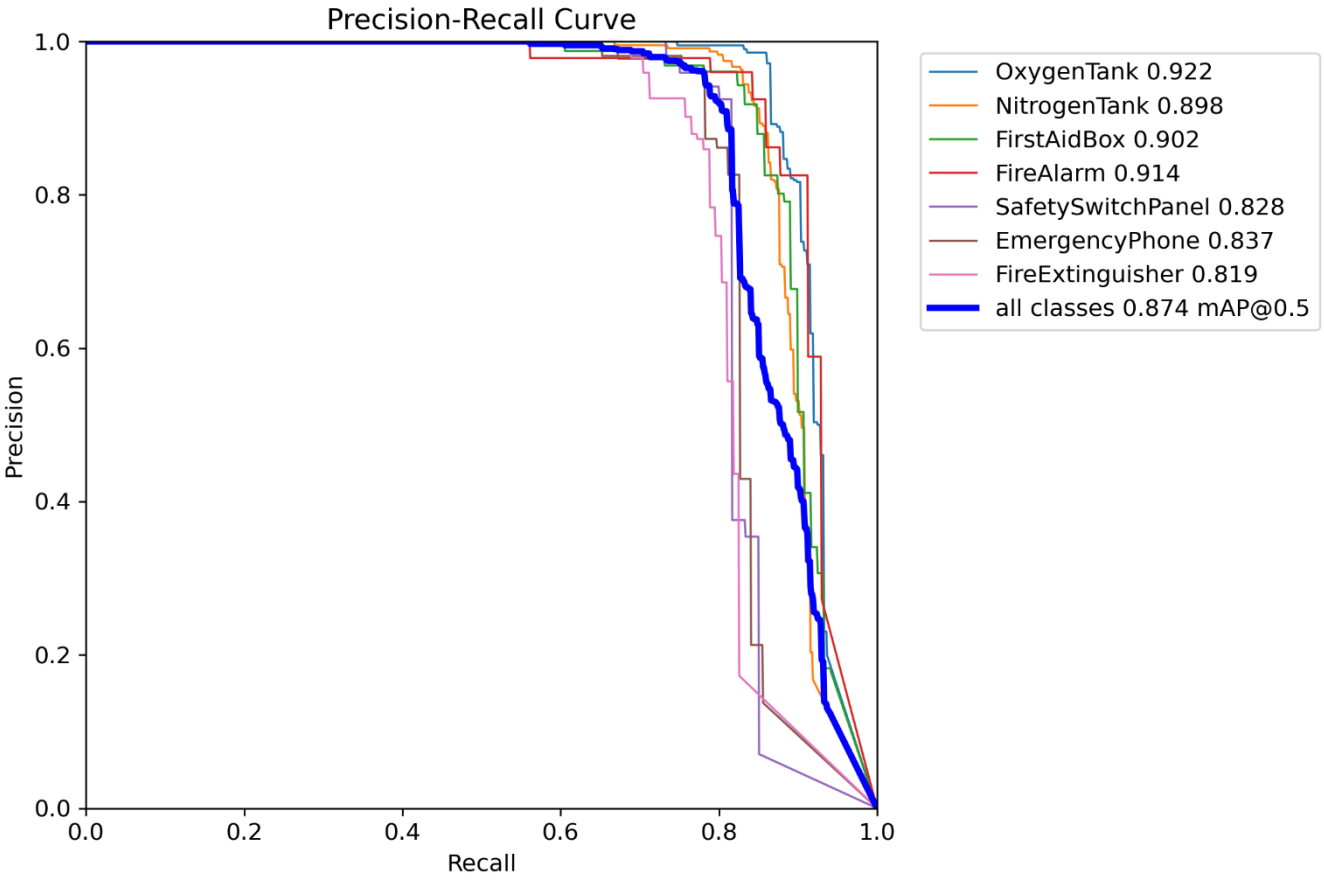
Performance Matrix and Confusion Matrix

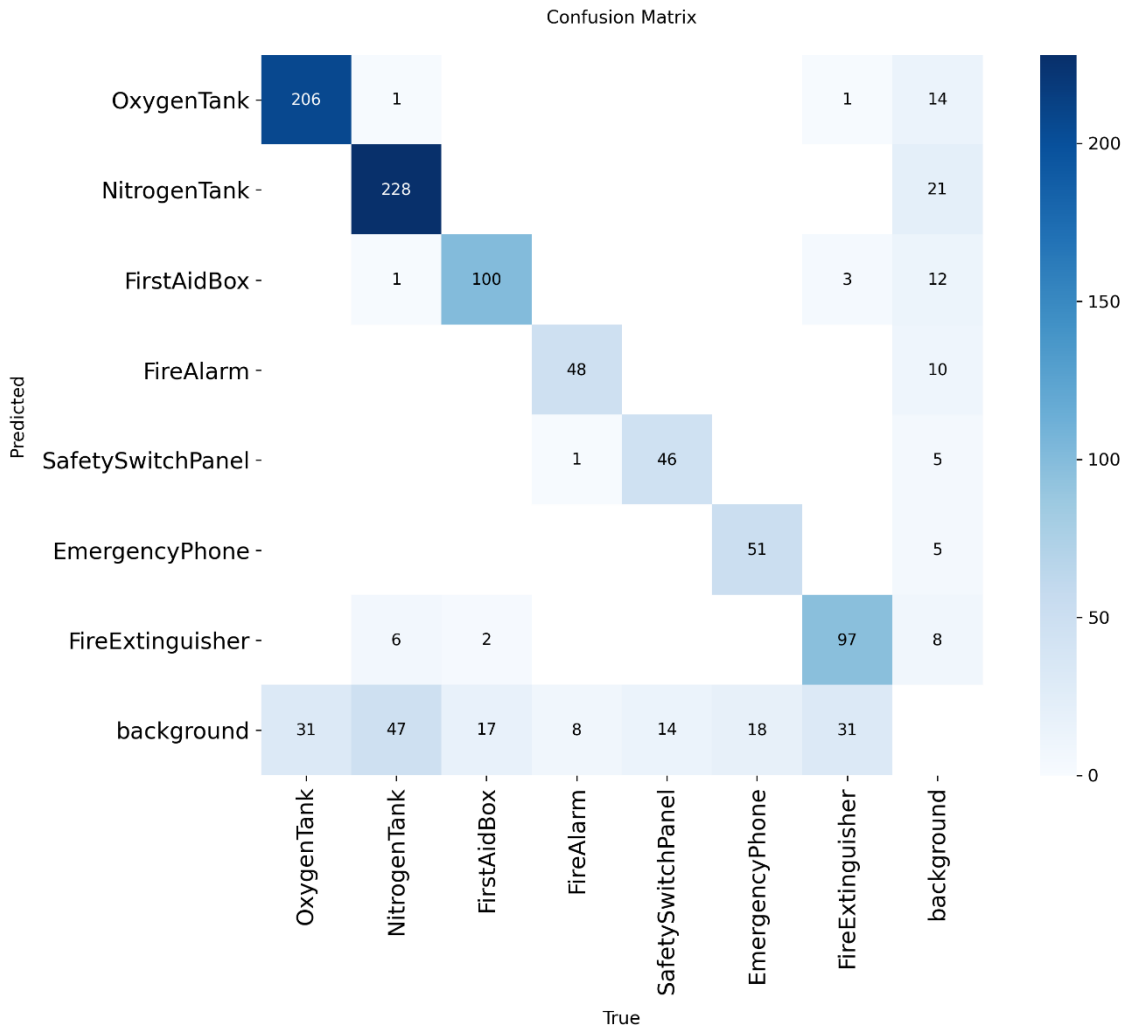
On Validation Set:

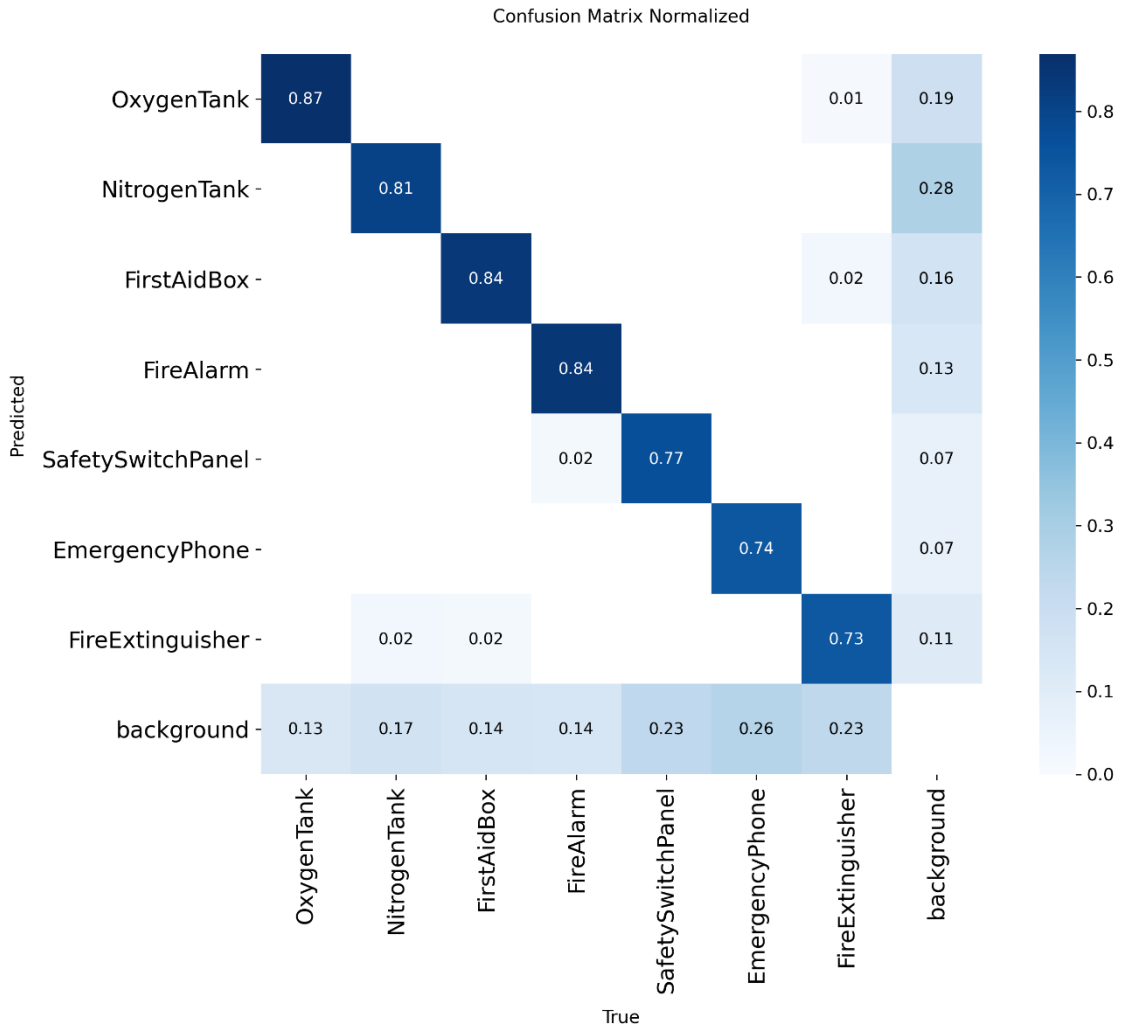



F1-Confidence Curve
F1
Confidence
0.0
0.2
0.4
0.6
0.8
1.0
OxygenTank
NitrogenTank
FirstAidBox
FireAlarm
SafetySwitchPanel
EmergencyPhone
FireExtinguisher
all classes 0.85 at 0.445

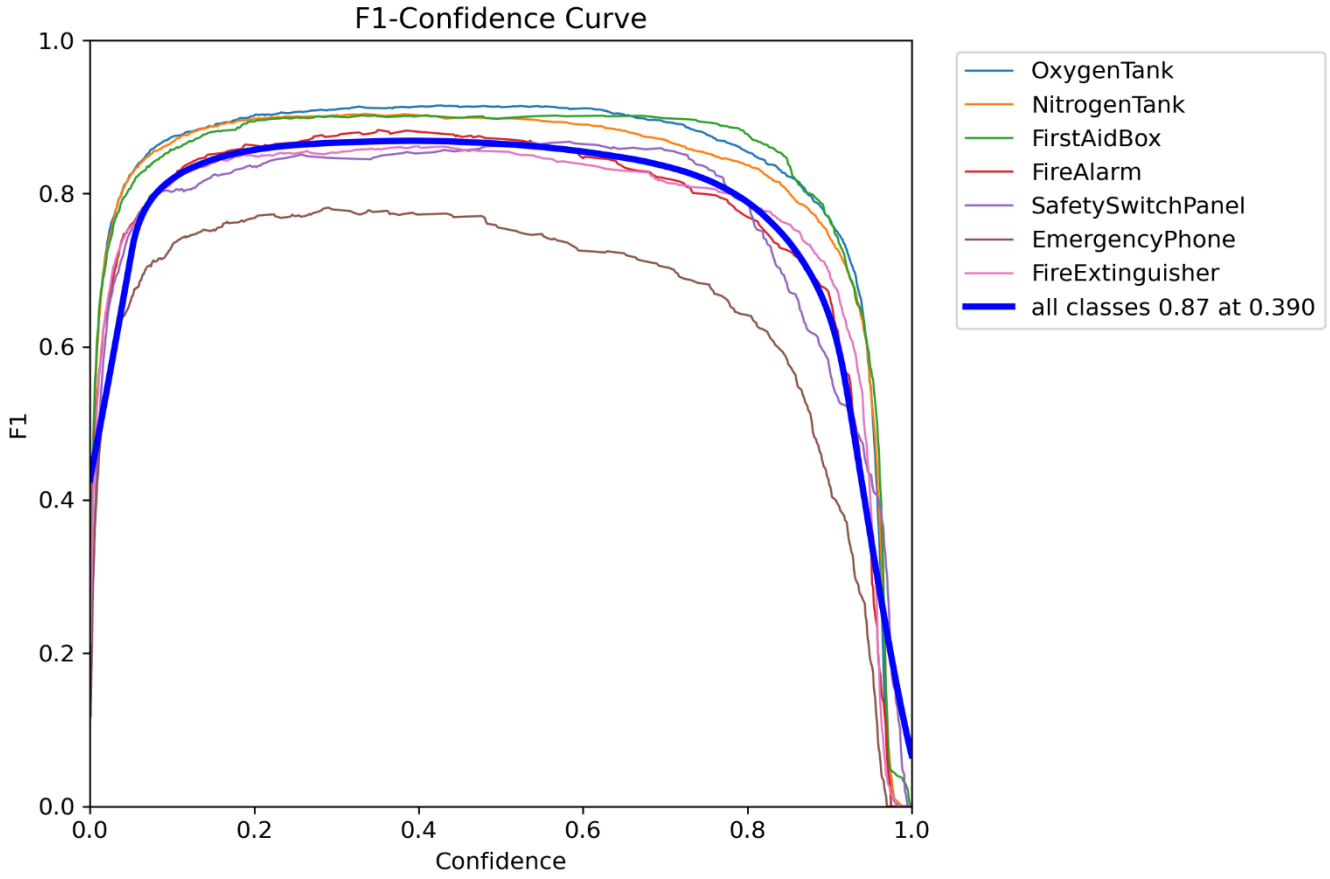

Precision-Confidence Curve
Precision
Confidence
0.0
0.2
0.4
0.6
0.8
1.0
OxygenTank
NitrogenTank
FirstAidBox
FireAlarm
SafetySwitchPanel
EmergencyPhone
FireExtinguisher
all classes 1.00 at 0.992

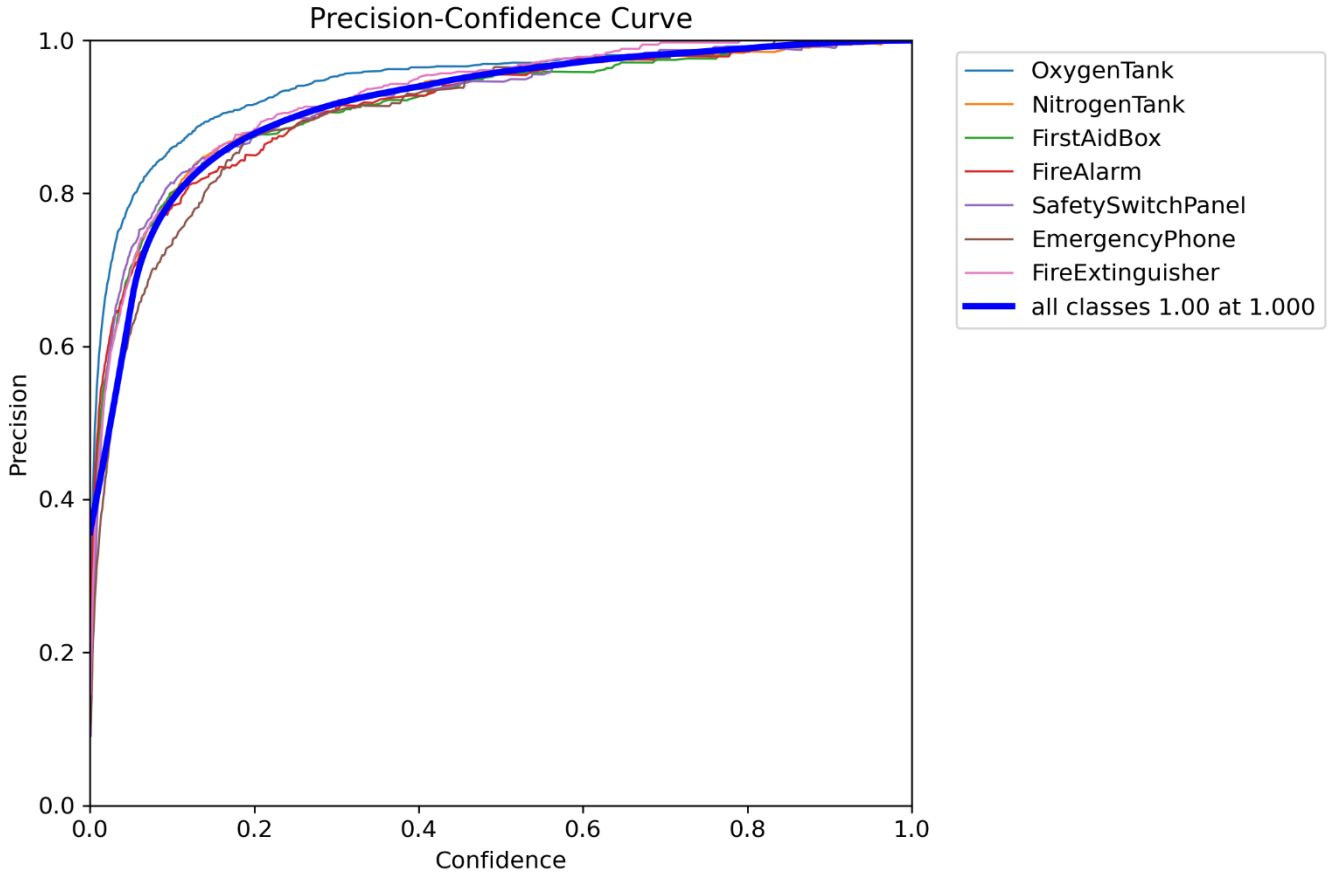

Precision-Recall Curve
Precision
Recall
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
OxygenTank 0.922
NitrogenTank 0.898
FirstAidBox 0.902
FireAlarm 0.914
SafetySwitchPanel 0.828
EmergencyPhone 0.837
FireExtinguisher 0.819
all classes 0.874 mAP@0.5

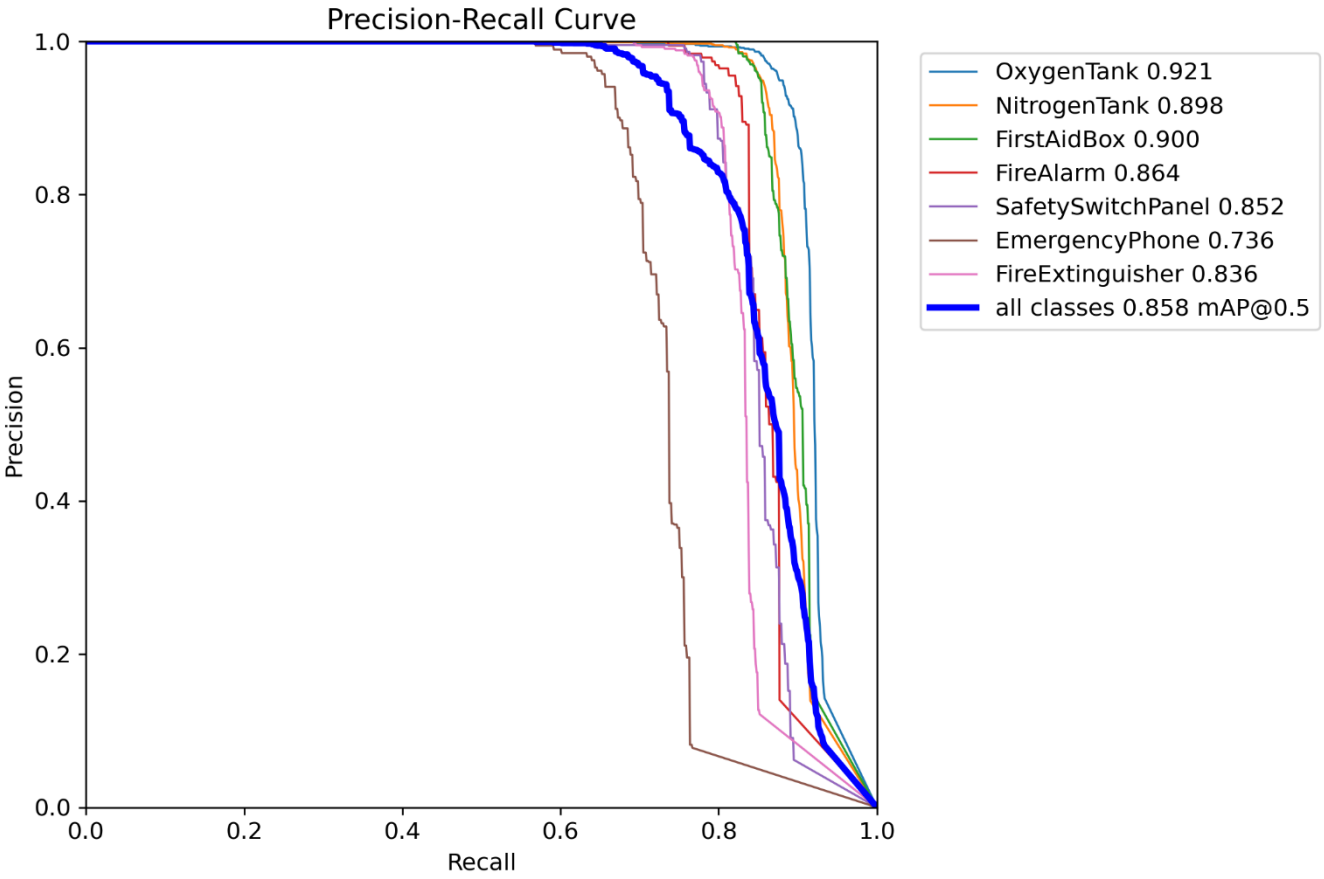

Confusion Matrix
OxygenTank
NitrogenTank
FirstAidBox
FireAlarm
SafetySwitchPanel
EmergencyPhone
FireExtinguisher
background
Predicted
True
206 1 1 14
228 21
1 100 3 12
48 10
1 46 5
51 5
6 2 97 8
31 47 17 8 14 18 31
200
150
100
50
0

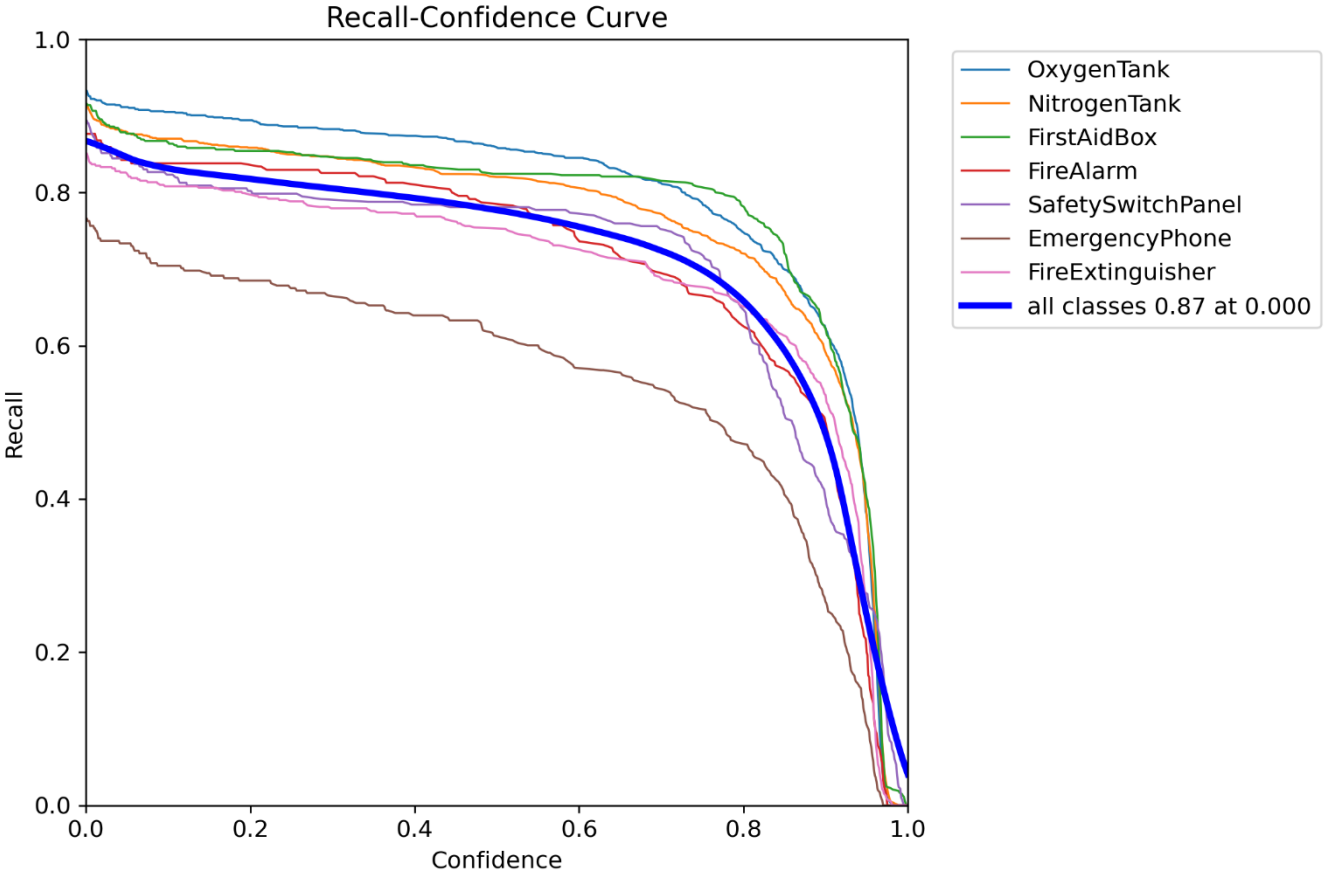

Confusion Matrix Normalized
OxygenTank
NitrogenTank
FirstAidBox
FireAlarm
SafetySwitchPanel
EmergencyPhone
FireExtinguisher
background
Predicted
True
0.87 0.01 0.19
0.81 0.28
0.84 0.02 0.16
0.84 0.13
0.02 0.77 0.07
0.74 0.07
0.02 0.02 0.73 0.11
0.13 0.17 0.14 0.14 0.23 0.26 0.23
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

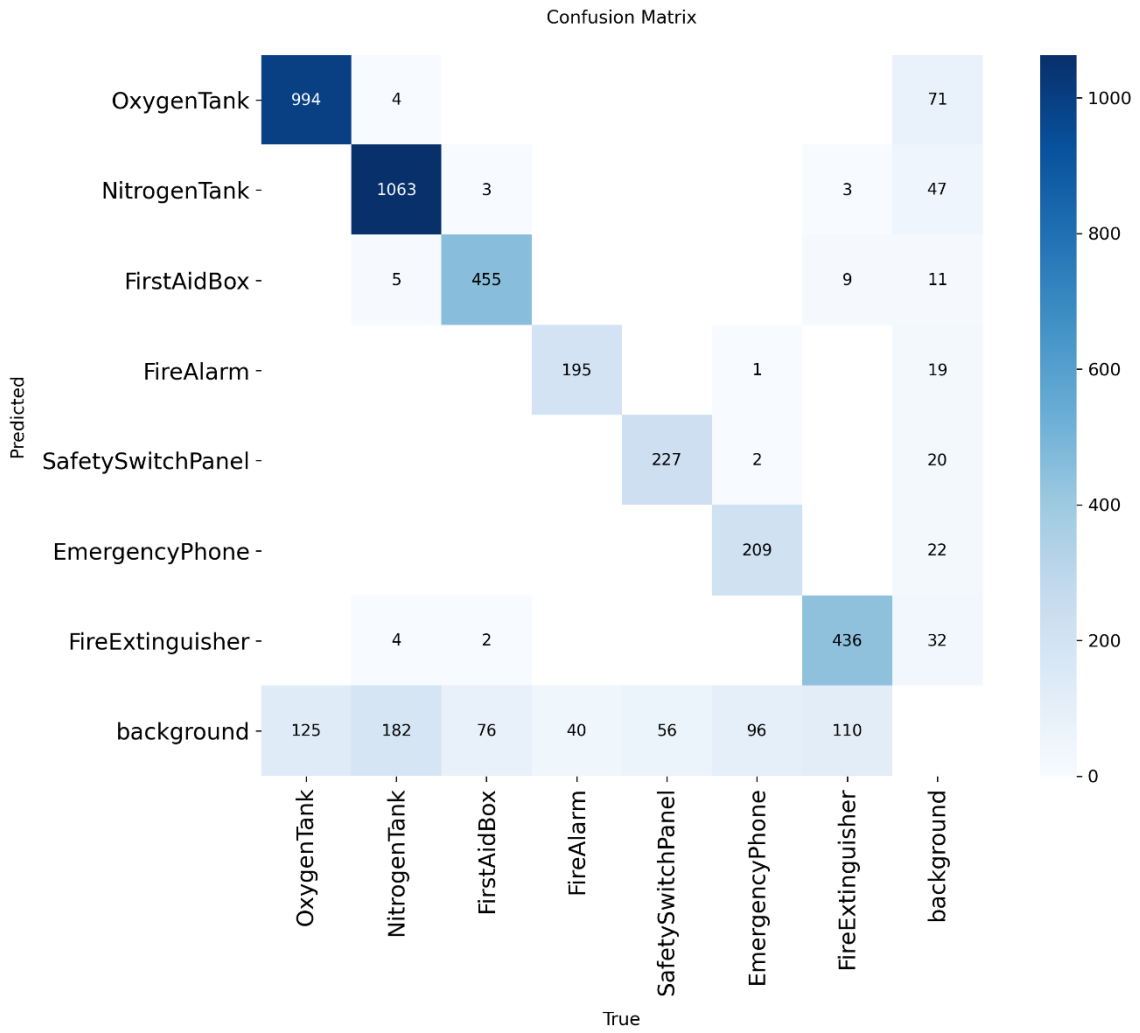
On Test Set:

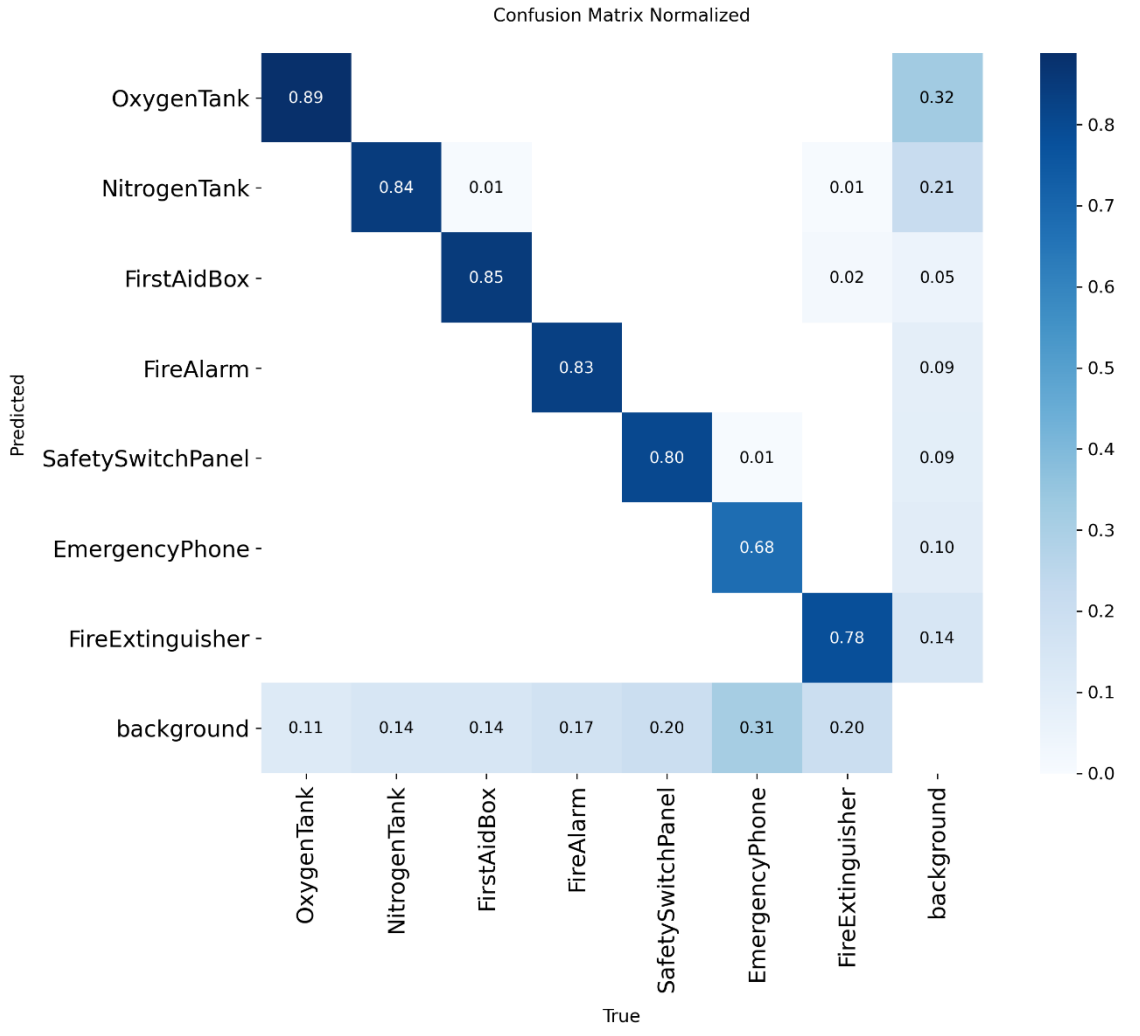

F1-Confidence Curve
F1
Confidence
0.0
0.2
0.4
0.6
0.8
1.0
OxygenTank
NitrogenTank
FirstAidBox
FireAlarm
SafetySwitchPanel
EmergencyPhone
FireExtinguisher
all classes 0.87 at 0.390


Precision-Confidence Curve
Precision
Confidence
0.0
0.2
0.4
0.6
0.8
1.0
OxygenTank
NitrogenTank
FirstAidBox
FireAlarm
SafetySwitchPanel
EmergencyPhone
FireExtinguisher
all classes 1.00 at 1.000

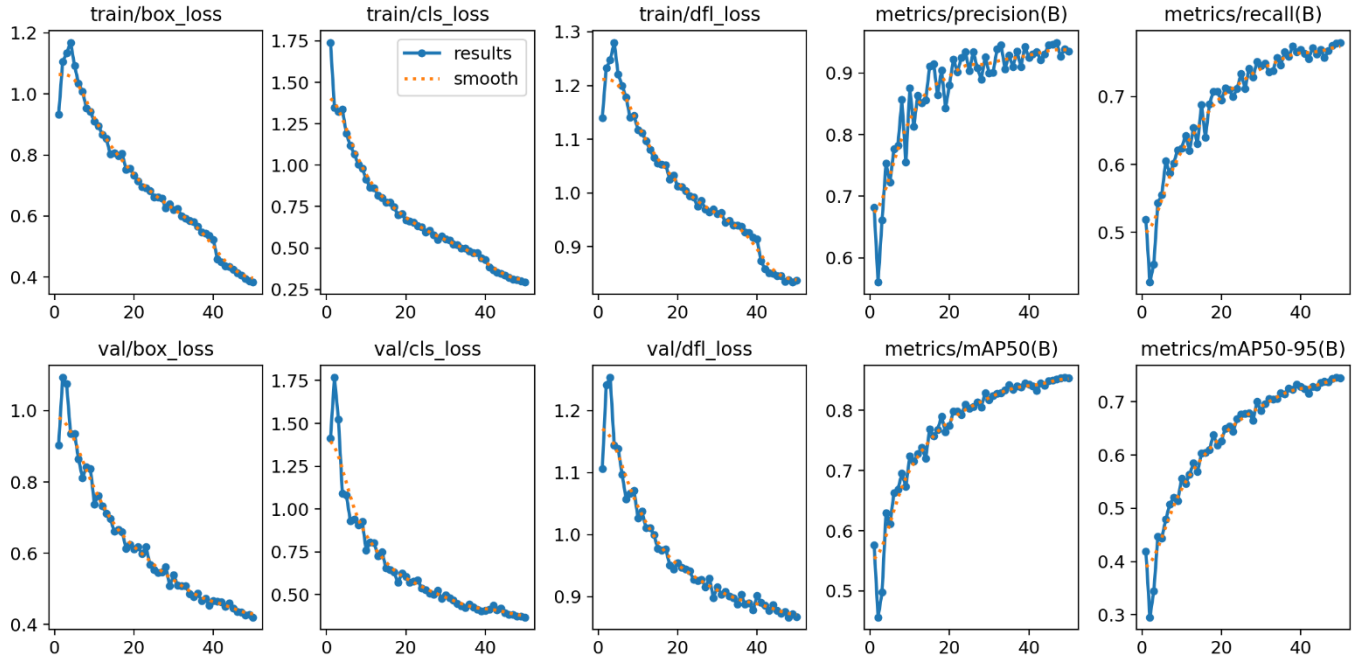

Precision-Recall Curve
Precision
Recall
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
OxygenTank 0.921
NitrogenTank 0.898
FirstAidBox 0.900
FireAlarm 0.864
SafetySwitchPanel 0.852
EmergencyPhone 0.736
FireExtinguisher 0.836
all classes 0.858 mAP@0.5


Recall-Confidence Curve
Recall
Confidence
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
OxygenTank
NitrogenTank
FirstAidBox
FireAlarm
SafetySwitchPanel
EmergencyPhone
FireExtinguisher
all classes 0.87 at 0.000


Confusion Matrix
OxygenTank
NitrogenTank
FirstAidBox
FireAlarm
SafetySwitchPanel
EmergencyPhone
FireExtinguisher
background
Predicted
True
994 4 71
1063 3 3 47
5 455 9 11
195 1 19
227 2 20
209 22
4 2 436 32
125 182 76 40 56 96 110
1000
800
600
400
200
0


Confusion Matrix Normalized
OxygenTank
NitrogenTank
FirstAidBox
FireAlarm
SafetySwitchPanel
EmergencyPhone
FireExtinguisher
background
Predicted
True
0.89 0.32
0.84 0.01 0.01 0.21
0.85 0.02 0.05
0.83 0.09
0.80 0.01 0.09
0.68 0.10
0.78 0.14
0.11 0.14 0.14 0.17 0.20 0.31 0.20
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

Results and Outputs



CHALLENGES AND SOLUTIONS

Challenge: Limited Computational Resources

- **Problem:** Training deep learning models requires significant GPU power, which is not always readily available.
- **Solution:** We utilized Google Colab's free T4 GPU resources. This allowed us to train a state-of-the-art model without any dependency on expensive local hardware, leveling the playing field and keeping the project cost-free.

Challenge: Achieving High Accuracy Rapidly

- **Problem:** Models often require extensive fine-tuning and long training cycles to reach high accuracy.
- **Solution:** The choice of YOLO11 was strategic. Its advanced architecture is designed for rapid convergence. As evidenced by our loss curves, the model learned features quickly and effectively, reaching a 85.6 mAP score without the need for prolonged hyperparameter tuning.

CONCLUSION AND FUTURE WORK

Conclusion

- Within the timespan, we successfully engineered a complete, high-performance object detection pipeline. By leveraging a powerful tech stack (YOLO11, Kaggle) and an efficient workflow, we progressed from raw data to a validated, high-fidelity model with a 85.6% mAP score. The project successfully demonstrates a viable and rapid path to developing production-ready computer vision models.

Future Work

- **Data Augmentation:** To improve robustness, we plan to implement data augmentation techniques (e.g., rotations, scaling, color shifts) to train the model on a more diverse dataset.
- **Hyperparameter Tuning:** While the default settings were highly effective, systematic hyperparameter tuning could yield further incremental improvements in accuracy.
- **Deployment & Optimization:** The next step is to deploy the model as a scalable API. For edge applications, we will explore optimization techniques like model pruning and quantization to reduce its size and inference time for deployment on low-power devices.