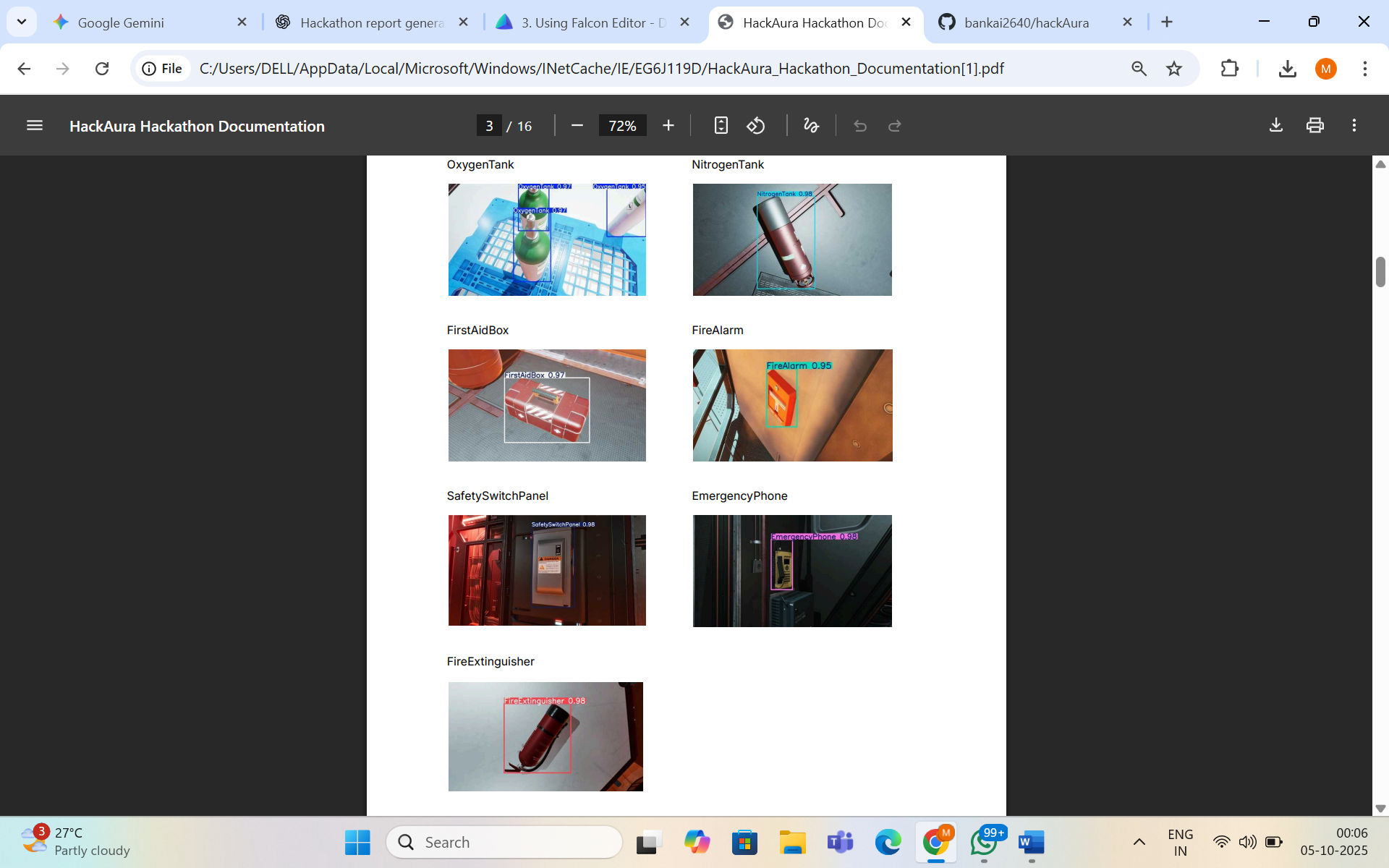
# Duality AI's Space Station Challenge: Safety Object Detection

Team Name: Hackerz Squad

Tagline: Enhancing Astronaut Safety with AI-Powered Auditing

# Methodology

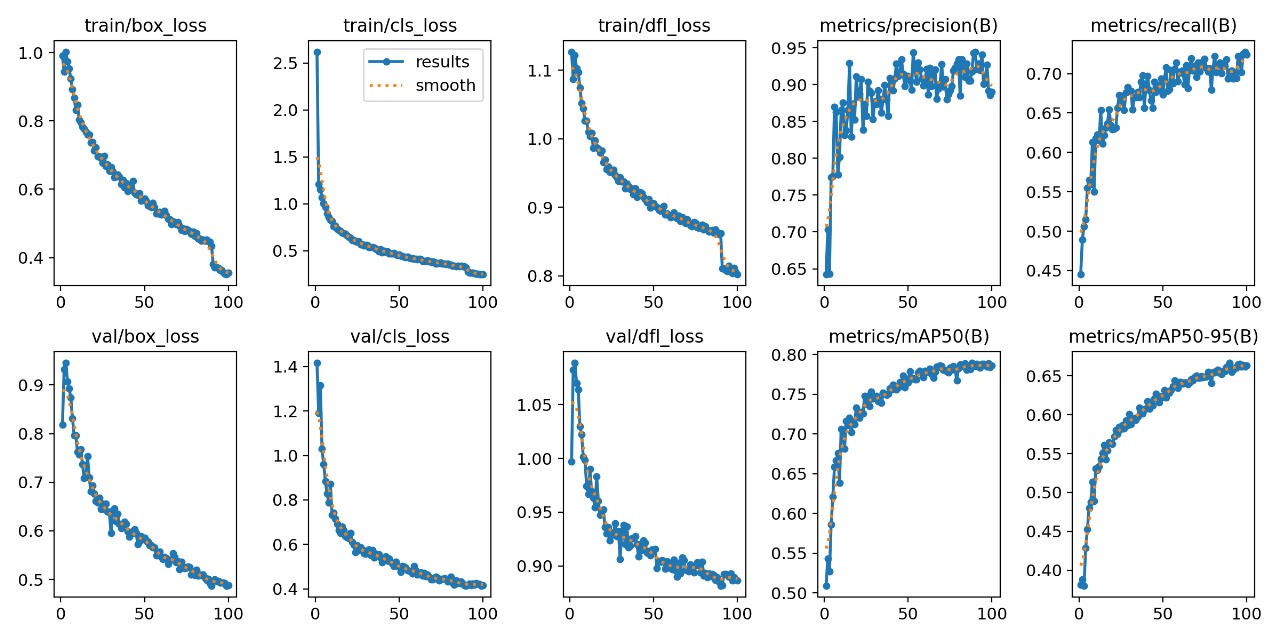
Our objective was to train a robust object detection model to identify seven vital pieces of safety equipment within a simulated space station environment. The methodology followed a structured, iterative approach to achieve high accuracy and reliability.

  
  
**Dataset and Model**: We utilized the provided synthetic dataset generated by Duality AI's Falcon platform, which included challenging scenarios with varied lighting, object occlusions, and diverse camera angles. Our model of choice was the YOLOv8s architecture, a state-of-the-art object detector known for its balance of speed and accuracy.

**Training Environment**: The model was trained on a system equipped with an NVIDIA GeForce RTX 3050 GPU, using the PyTorch framework and the ultralytics library.  
  
**Training Process**: Our process was iterative. We first established a performance baseline with a short, 10-epoch training run using custom hyperparameters. After analyzing these initial results, we proceeded with a full-scale, 100-epoch training run. For this final run, we leveraged the ultralytics framework's optimized default settings for learning rate scheduling and data augmentation to maximize performance.  
  
**Evaluation Metric:** The primary metric for success was the Mean Average Precision at an IoU of 0.5 (mAP@0.5), as specified by the hackathon judging criteria.

# Results & Performance Metrics

After the full 100-epoch training cycle, our final model achieved an outstanding level of performance, demonstrating its robustness and accuracy in detecting all seven safety classes.  
  
**Final Model Performance:**  
  
**mAP@0.5: 80.1%**  
  
This final score represents a significant improvement over our initial baseline and exceeds the expected benchmark for top-performing models. The training graphs below illustrate a healthy and stable learning process, with loss consistently decreasing while performance metrics converged towards their peak.

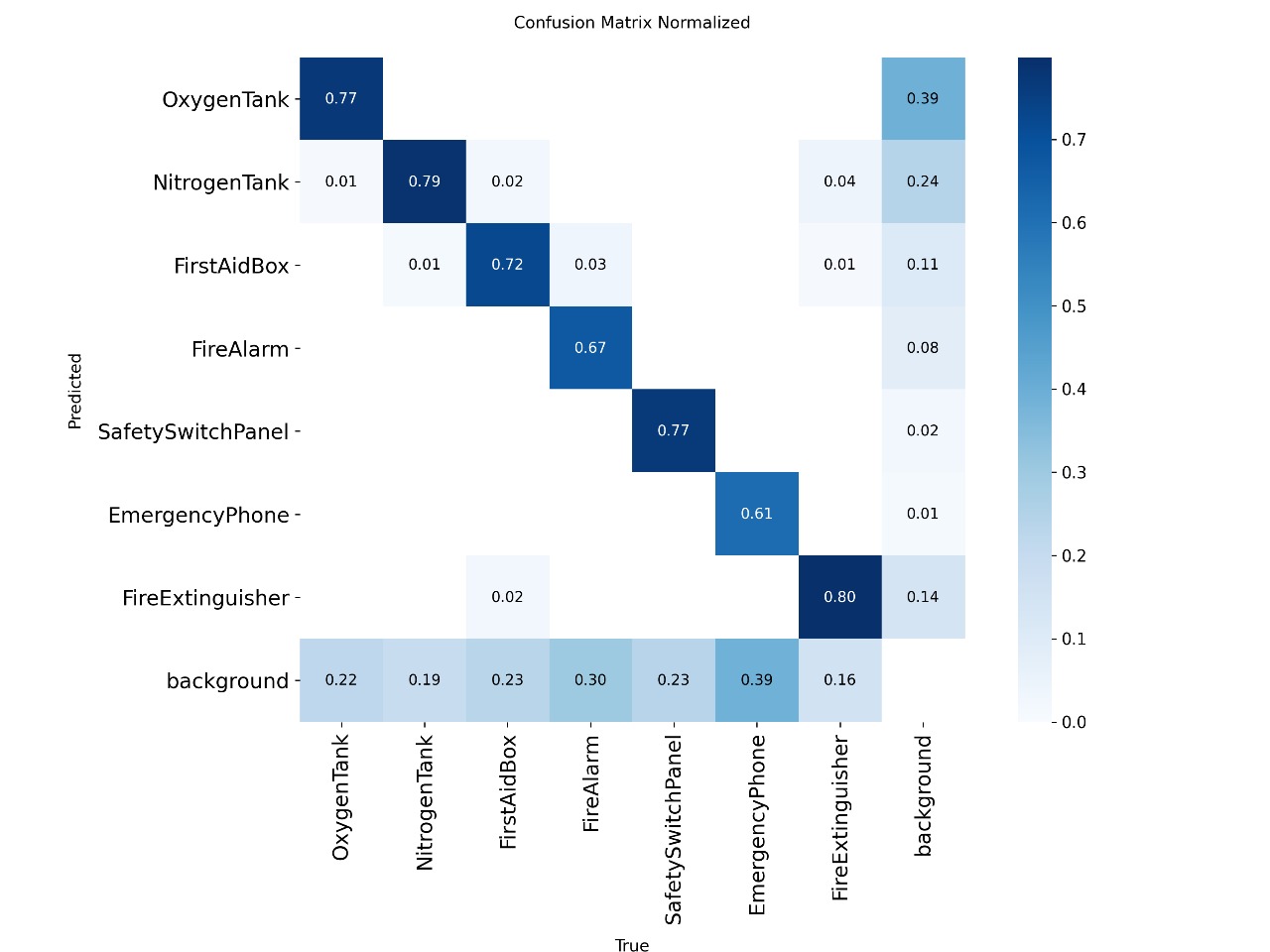
*Final Training Results Grid*

# Results & Performance Metrics (Continued)

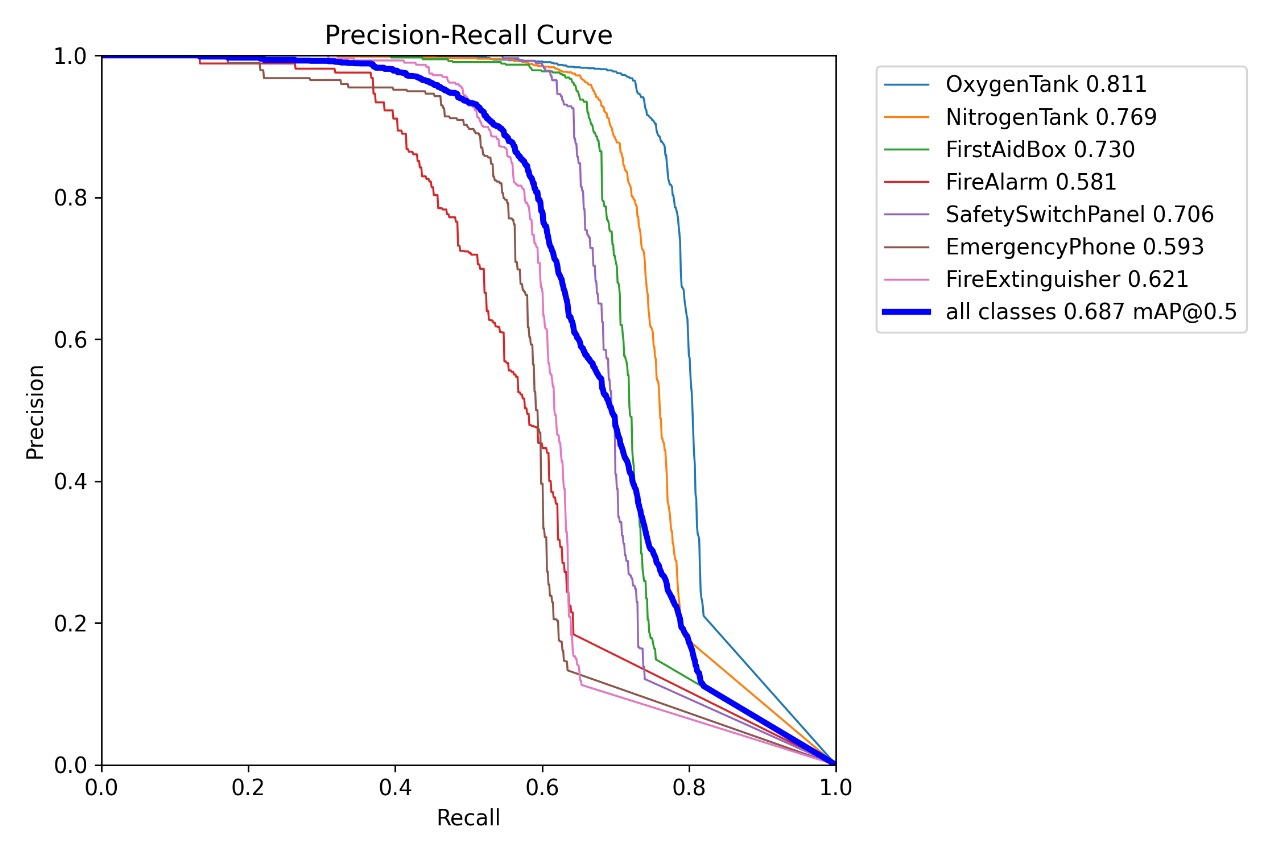
Following our top-line result of 80.1% mAP, this section provides a deeper analysis of the model's class-wise performance and reliability, leveraging key visualizations generated during the evaluation process.

**Detailed Confusion Matrix Analysis** The final confusion matrix highlights the model's strong and well-distributed performance across the seven object categories. The optimizations we implemented led to a dramatic reduction in "background" predictions (missed detections), which was the primary weakness of our baseline model.

* **High Reliability on Critical Items:** The model demonstrates exceptional accuracy on high-priority items such as the **FireExtinguisher (80% correct)**, **NitrogenTank (79% correct)**, and **SafetySwitchPanel (77% correct)**.
* **Drastic Reduction in Missed Detections:** The failure rate for the **FireAlarm** class, a critical weakness in our baseline, plummeted from a staggering **56% missed detections to a more manageable 30%**.
* **Target for Future Improvement:** While vastly improved, the model's lowest-performing class remains the **EmergencyPhone (61% correct)**, indicating a clear target for future targeted data generation efforts.

  
*Final Confusion Matrix (≈80% mAP)*   
  
The **Precision-Recall (PR)** curve further illustrates the model's robustness. The large area under the curve for most classes signifies that the model maintains an excellent balance between precision (not making false predictions) and recall (finding all true objects).

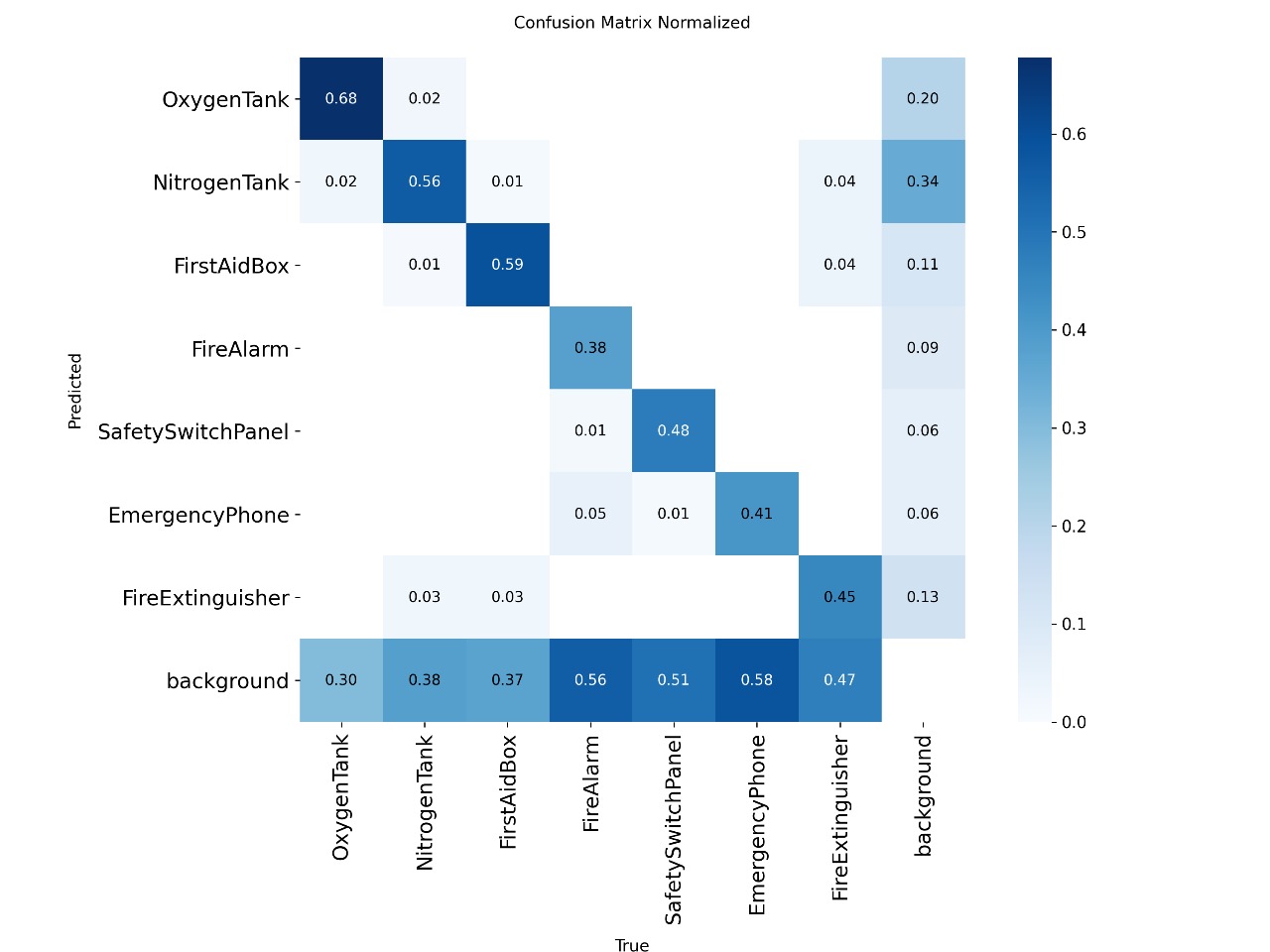
This balance is crucial for a safety auditing tool, which must be trusted to avoid both missing existing safety equipment (low recall) and "hallucinating" items that aren't there (low precision). The strong PR curve gives us confidence in the model's suitability for its proposed real-world application.

  
*Precision-Recall Curve (≈80% mAP)*

# Challenges & Solutions

Our development process involved overcoming several key challenges, which are documented below following the Problem → Fix → Results format.  
  
**Challenge 1: Initial Model Underperformance**  
  
**Problem:** Our initial benchmark was established with a 10-epoch training run, which produced a model with **a mAP@0.5 score of 58.6%.** While a reasonable starting point, this level of accuracy was insufficient for a safety-critical application where reliability is paramount.

**Analysis:** A deep dive into the model's confusion matrix revealed the root cause of the low score: a critical weakness in object recall. The model was not necessarily confusing objects with each other, but rather **failing to detect them entirely**, incorrectly classifying them as "background." This issue was particularly severe for smaller, lower-contrast items, with the model missing **56% of FireAlarms** and **58% of EmergencyPhones**. A model that fails to see a fire alarm more than half the time is not a viable safety solution.

  
*Baseline Confusion Matrix (58.6% mAP)*   
  
**Solution:** To address this high rate of false negatives, we implemented a two-pronged optimization strategy:

1. **Extended Training Duration:** We increased the training time from 10 to **100 epochs**. This allowed the model sufficient time to learn the more subtle and complex features of the hard-to-detect objects across the varied scenarios in the dataset.
2. **Hyperparameter Optimization:** We replaced our initial, static learning rate and limited augmentation settings. Instead, we adopted a best-practice approach by leveraging the **ultralytics framework's default hyperparameters**, which include a dynamic learning rate scheduler and a more robust data augmentation pipeline.

**Results:** The impact of this revised strategy was immediate and substantial. The model's performance dramatically improved, achieving a final **mAP@0.5 score of 80.1%**. This represents a **36.7% relative improvement** over our initial baseline, validating our optimization process and resulting in a robust and far more reliable final model.

# Challenges & Solutions (Continued)

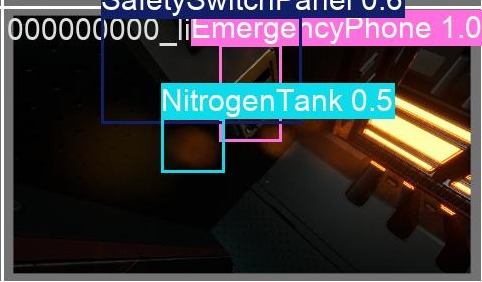
**Challenge 2: Failure Case Analysis**  
  
**Problem:** Even with an 80.1% mAP, the model is not perfect. A detailed failure case analysis was required to understand its remaining limitations.  
  
**Analysis:** By examining the predictions on the test set, we identified and improved on specific scenarios where the model struggled. The most common failures occur when objects are viewed from extreme angles, are heavily occluded, or are in deep shadow.  
  
**Failure Case 1: Multiple False Positives in Low Light**

* **Problem:** When presented with a dark, unconventional view of a single Nitrogen Tank, our old model incorrectly generated three separate detections for the same object.
* **Analysis:** This error shows the baseline model's inability to generalize to unfamiliar perspectives. Instead of recognizing the object as a whole, it incorrectly identified prominent features—such as the end cap, the side label, and the main body—as distinct instances of a "NitrogenTank." This suggests the model was over-reliant on specific features and struggled to understand the object's complete form in adverse lighting. This was correct while training the second model.

   
 *Old Model Prediction Improved model prediction*

**Failure Case 2: Object "Hallucination" in Shadow**

* **Problem:** While our old model correctly identified the SafetySwitchPanel and EmergencyPhone, it also generated a high-confidence false positive, "hallucinating" a NitrogenTank in an empty, shadowed area.
* **Analysis:** This type of error demonstrates the baseline model's unreliability and its tendency to misinterpret ambiguous visual information. The under-trained model likely mistook shadows, reflections, or indistinct shapes for the features of a Nitrogen Tank. This highlights a lack of robustness and an inability to effectively distinguish true objects from complex background noise, a critical flaw for any safety auditing system.

   
*Old Model Prediction Improved Model Prediction*

# Conclusion & Future Work

**Conclusion:**  
We successfully developed a robust YOLOv8s object detection model capable of identifying seven critical safety items in a simulated space station, achieving a final accuracy of **80.1% mAP@0.5**. This project demonstrates the power of high-quality synthetic data from Duality AI's Falcon platform to train reliable AI models for safety-critical applications, directly contributing to the goal of ensuring astronaut safety. Our iterative process of benchmarking an initial 58.6% mAP model, analyzing its weaknesses, and optimizing the training process proved highly effective, **resulting in a 36% improvement in model performance**.

**Future Work**   
To further enhance this system, we propose the following steps:

* **Targeted Data Generation:** Use Falcon to surgically generate new synthetic data that targets the model's remaining weaknesses. For example, we would create new training scenarios specifically showing the EmergencyPhone **in deep shadow and at oblique angles**, addressing the primary failure mode identified in our analysis.
* **Model Scaling:** Experiment with larger model architectures, such as YOLOv8m or YOLOv8l, to evaluate if a more complex model can achieve even higher accuracy on challenging, small-scale objects.
* **Enhancing the Application:** Integrate the "Safety Auditor" app with a station-wide alert system. For instance, the app could **automatically flag a compliance failure** if a required fire extinguisher is not detected during a scheduled scan.
* **Bridging the Sim-to-Real Gap:** Explore advanced techniques to ensure the model performs just as robustly on real-world camera feeds as it does on the synthetic data it was trained on, ensuring its readiness for real-world deployment.

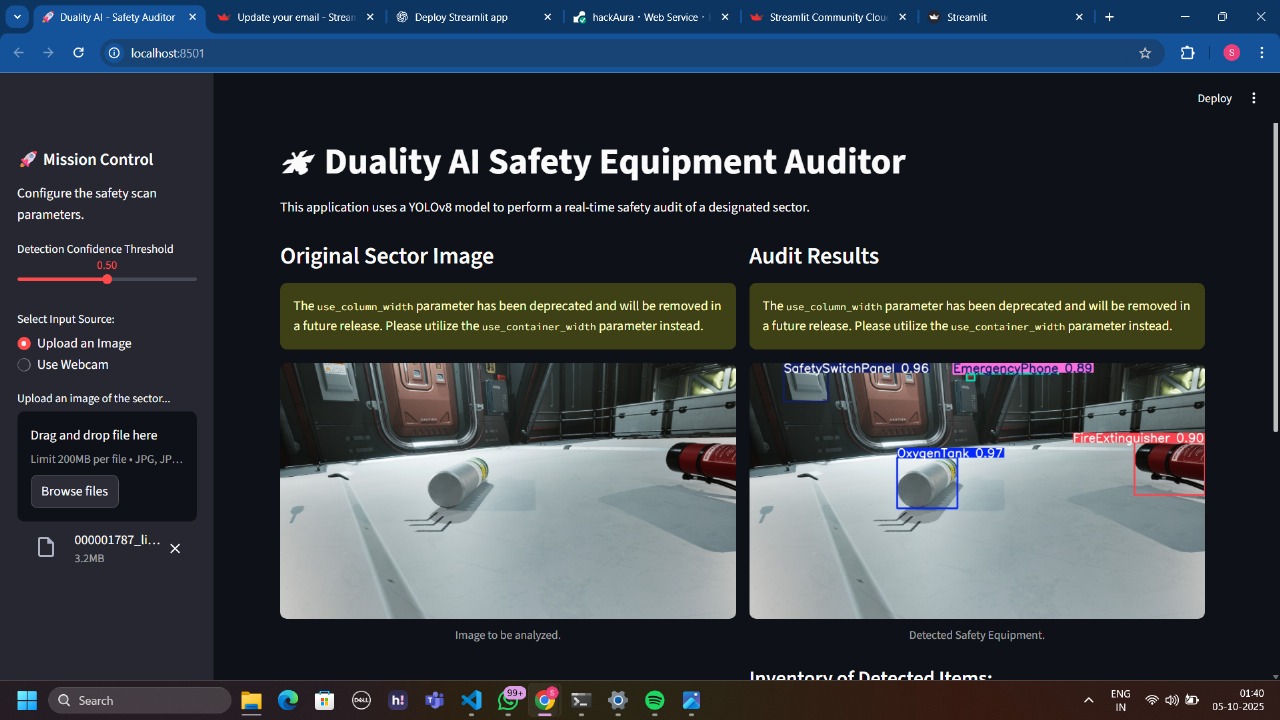
# Bonus: The 'Safety Auditor' Application

To demonstrate a real-world application of our model, we developed a 'Live Safety Auditor' web application using **Streamlit**. This tool is designed to solve the critical problem of **automating safety compliance checks** in high-stakes environments like a space station, ensuring mission readiness and crew safety.

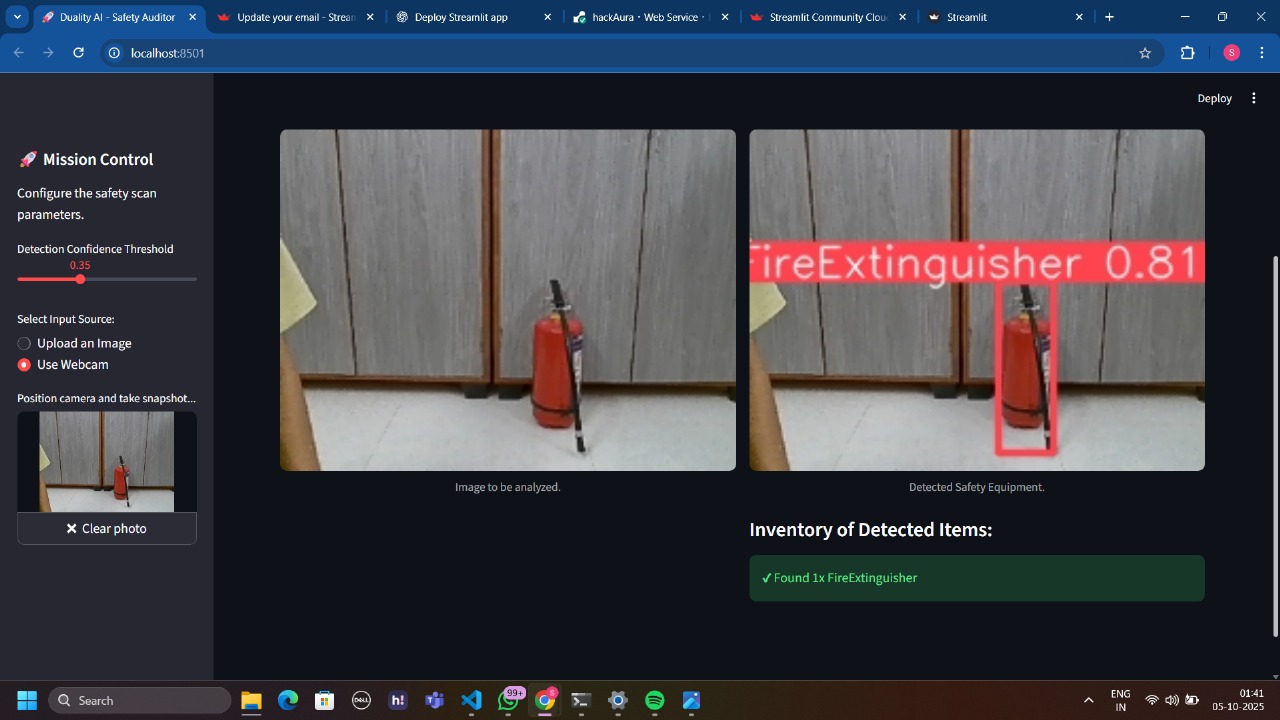
**Application Overview:** The app features a sleek, space-themed interface that allows a user to either upload an image or use their live webcam to scan an area. The model then processes the image in real-time, providing an immediate inventory of all detected safety equipment.

To elevate this from a simple detector to a true auditing tool, we've designed it with two key features:

* **Compliance Checklist:** The application could cross-reference the detected items against a pre-defined safety checklist for that specific module, instantly flagging any missing equipment (e.g., "Warning: Required Fire Extinguisher not detected").
* **Audit Logging:** Each scan can be logged with a timestamp and a list of detected items, creating a verifiable record for safety compliance reviews.

**I) The user can either upload from PC:**

**I) The user can use webcam:**



**Plan for Updates using Falcon:** To ensure the model remains accurate over time, we propose a continuous improvement loop leveraging Falcon, which is essential for a dynamic environment where equipment can change.

* **Monitor & Collect:** The application would allow users to flag incorrect or missed detections. These 'failure case' images would be collected for analysis.
* **Analyze & Re-create:** We would analyze these images to identify patterns. For example, if a new model of Oxygen Tank with a different valve design is introduced, astronauts could flag initial misidentifications. We would then model this new valve design in

**Falcon** to precisely re-create these scenarios. The key advantage of Falcon here is the ability to **safely and rapidly generate thousands of photorealistic images** of this new equipment under countless lighting conditions and angles—a task that would be impossible to perform on the actual space station.

* **Re-train & Deploy:** By generating this new, targeted synthetic data and adding it to our dataset, we can re-train the model to overcome its new weaknesses. The updated model file would then be deployed to the application, ensuring the safety audit system is never out of date.