Eleven11: Spot. Detect. Protect.

AI for detecting essential safety gear

Team Name: 11:11 Crew

Project Objective:

Our team is called 11:11 Crew, inspired by the idea that 11:11 symbolizes perfect timing, clarity, and intention. We believe this reflects our goal: creating an AI system that works just when it's needed most. Our project, Eleven11, is an object detection web app designed to identify critical safety tools in space environments. By using computer vision, our goal is to help astronauts and technicians locate items like oxygen tanks, fire extinguishers, and toolkits quickly and reliably.

1. Methodology

Think of building our AI model like teaching a very smart student. We followed a step-by-step process to ensure it was learned effectively.

Gathering Our Learning Material (Dataset Preparation)

- We collected a dataset of images containing fire extinguishers, toolkits, and oxygen cylinders in realistic environments.
- For every image, we manually drew boxes around each item. This is similar to showing flashcards repeatedly and saying, "This is a fire extinguisher. This is a toolkit."
- We divided the dataset into:
    - A training set to teach AI.
    - A validation set to check if the AI is learning correctly.
    - A test set to evaluate how well it performs on unseen data.

Model Selection and Initialization

- We chose the YOLOv8-Large (`yolov8l.pt`) model because it's fast, accurate, and capable of learning detailed patterns in images.
- The model was initialized with pretrained weights and adapted to detect our 3 custom object classes.

Training Configuration

- We trained the model for 50 epochs on Google Colab using a Tesla T4 GPU.
- All images were resized to 640×640 pixels.
- A batch size of 8 was used, meaning weights were updated every 8 images.
- Early stopping was enabled with a patience of 15 to stop training when improvements slowed.

Optimization and Regularization

To help the model learn better and generalize to new images:

- **Optimizer:** Stochastic Gradient Descent (SGD) with an initial learning rate of 0.001.
- **Geometric Augmentations:** Slight image rotations, zooming in/out (scale = 0.8), and flips (horizontal = 50%, vertical = 10%).
- **Color Adjustments:** Changed hue, saturation, and brightness to teach the model about lighting variations.
- **Advanced Augmentations:**
  - **Mosaic:** Merged four images into one to simulate clutter.
  - **MixUp:** Combined two images to create complex examples**.**

2. How Well It Works (Results & Performance Metrics)

Our model improved significantly over time and became highly reliable.

Overall Performance Scores

- **mAP@0.5: 0.9766 (97.66%)**
  This score tells us how accurately the AI finds each object and draws the box around it.

- **mAP@0.5:0.95: 0.9430 (94.30%)**
  This is a stricter version of mAP, measuring how precise the boxes are under varying thresholds. Our model did very well.

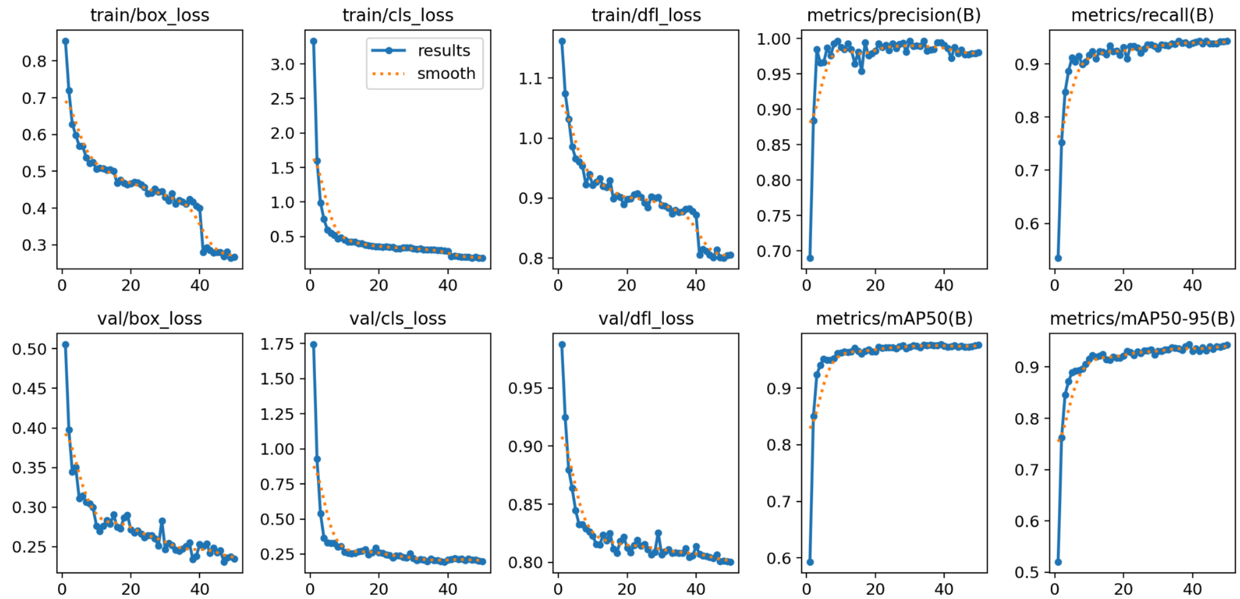
- **Precision: 0.9801 (98.01%)**
  When our AI says "I found a fire extinguisher," it's right nearly 98% of the time.

- **Recall: 0.9429 (94.29%)**
  Out of all the actual safety tools in the image, our AI finds 94% of them. This shows it misses very few items.

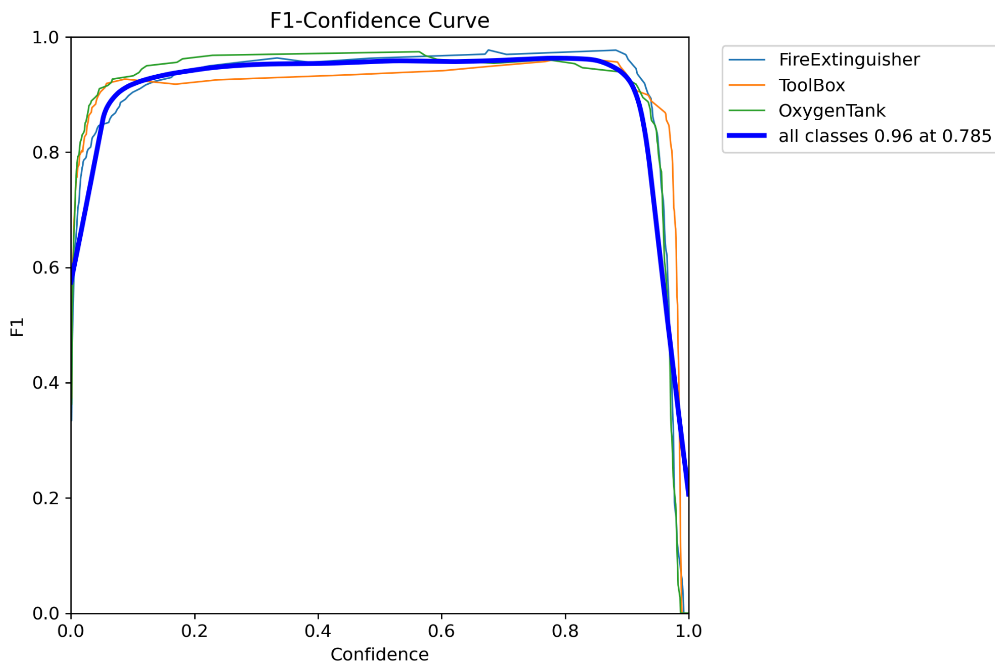
Visualizing Model Performance

The following visualizations will be included in the final report:

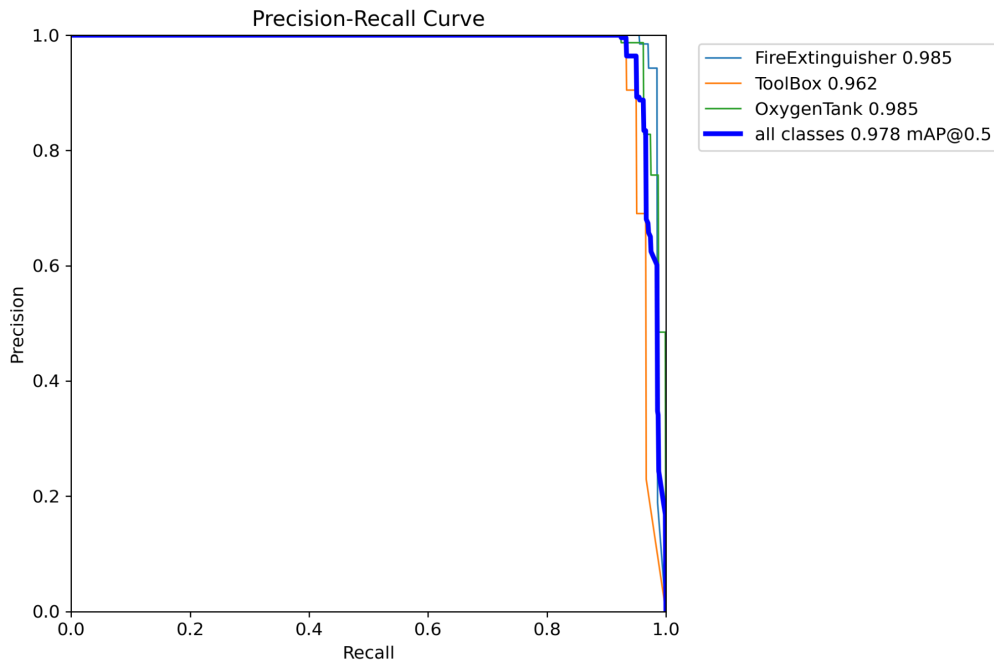
- Training loss curves (results.jpg)



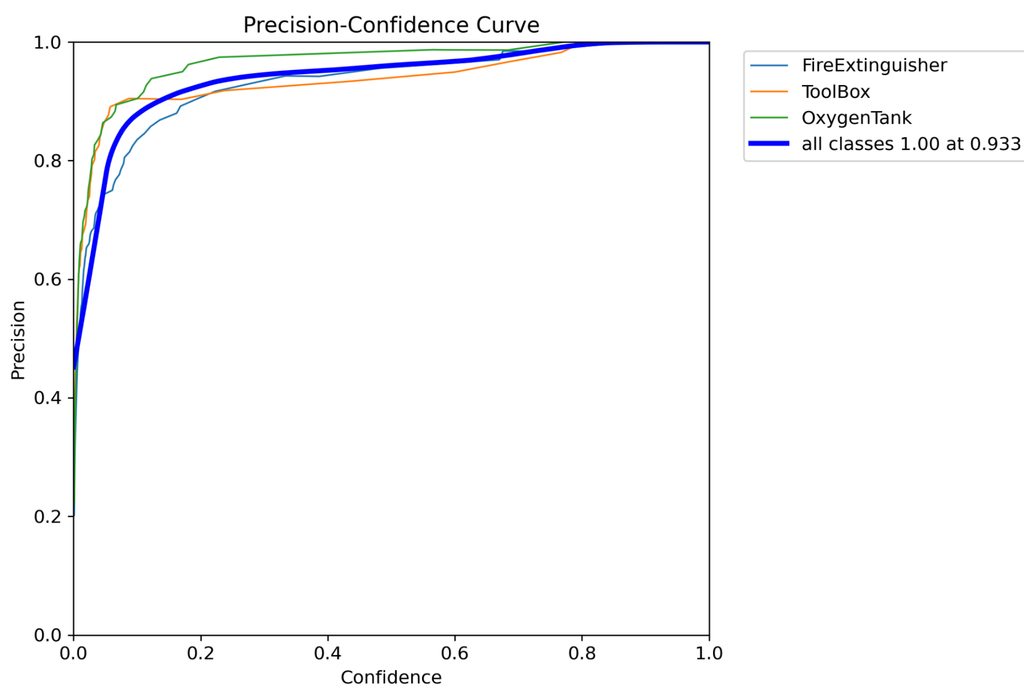
- F1-score curve (F1_curve.png)



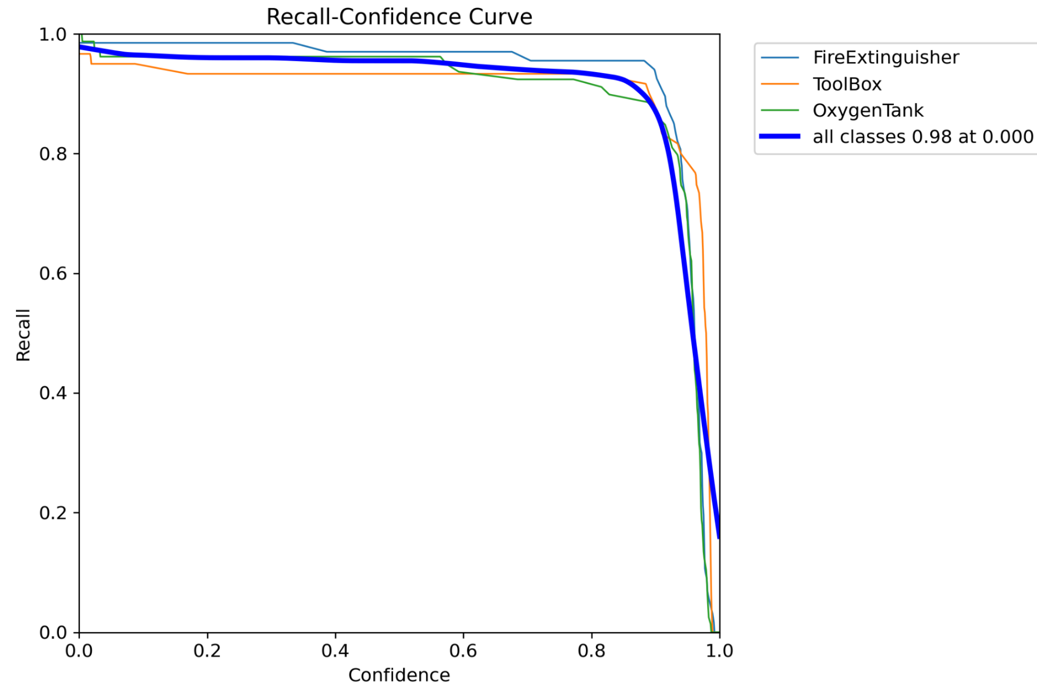
- Precision-Recall curve (PR_curve.png)



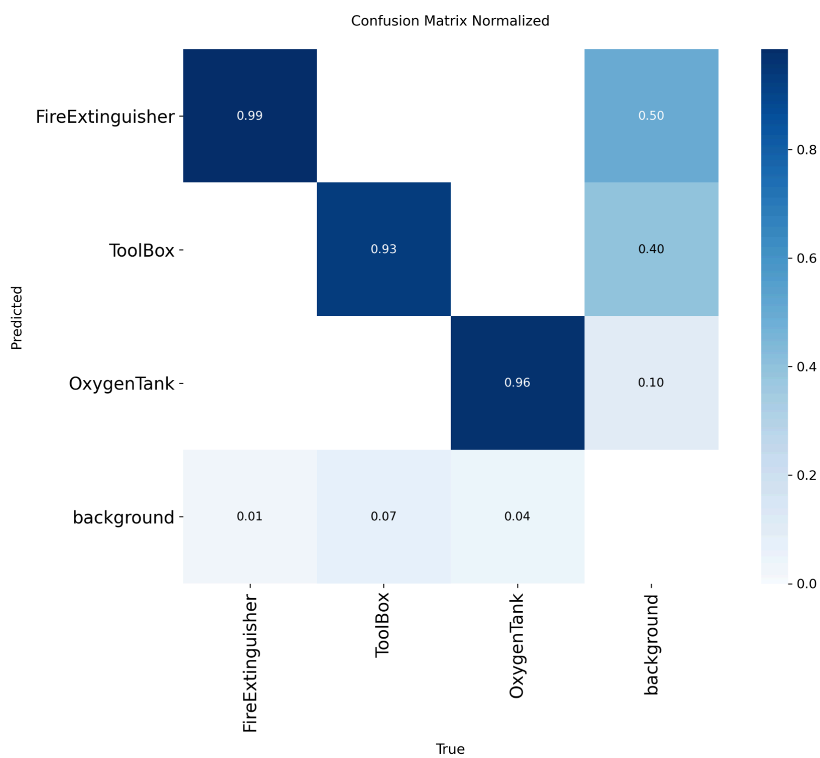
- Precision vs Confidence curve (P_curve.png)



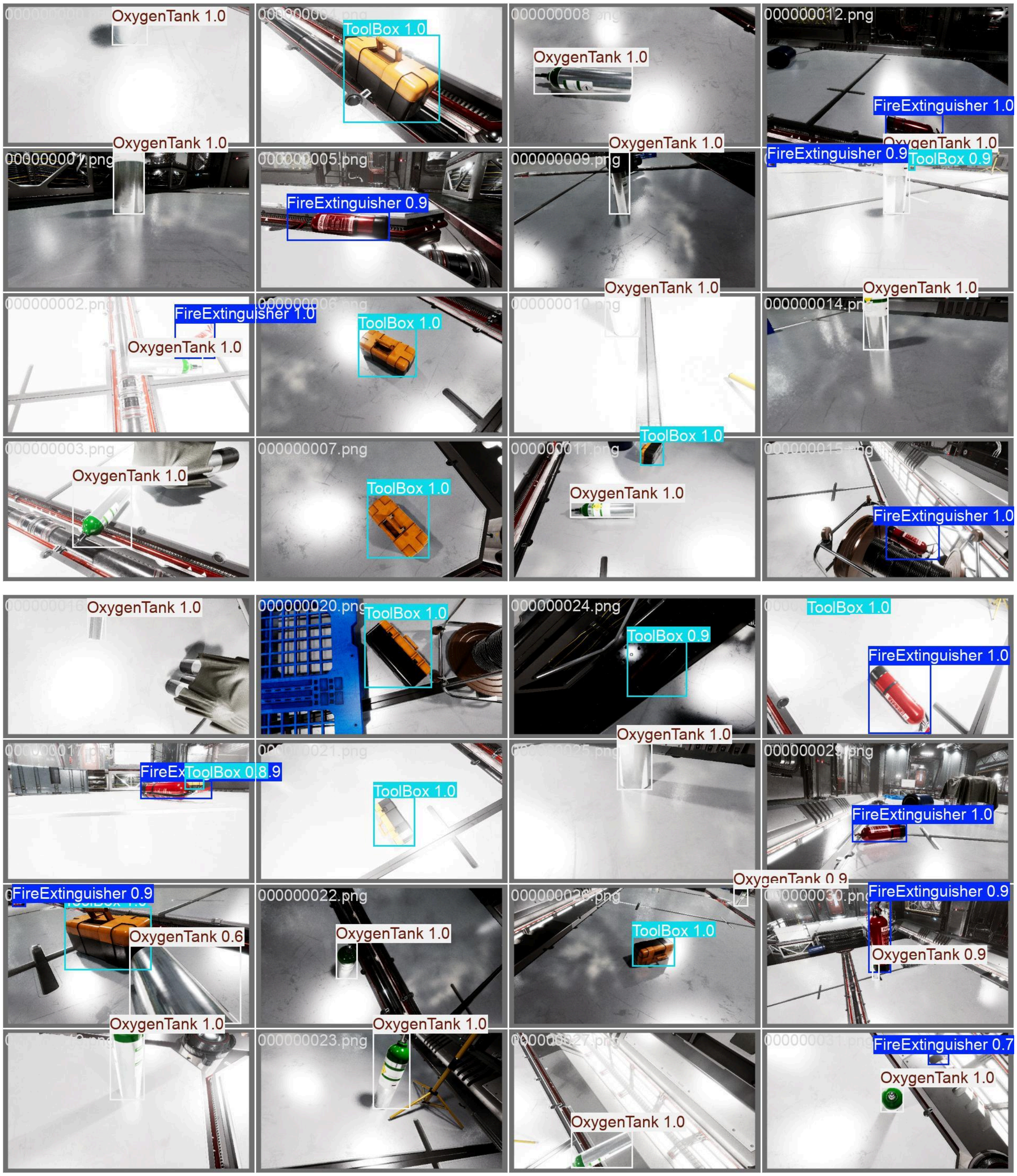
- Recall vs Confidence curve (R_curve.png)



- Confusion Matrix



- Sample detection outputs from the test set



These plots illustrate the training progress, detection quality, and class-specific performance.

Per-Class Performance

The model's performance for each safety item class is summarized below:

- **Fire Extinguisher**
  - **Precision:** 0.992
  - **Recall:** 0.955
  - **mAP@0.5:** 0.985
  - **mAP@0.5:0.95:** 0.941

- **Toolbox**
  - **Precision:** 0.991
  - **Recall:** 0.933
  - **mAP@0.5:** 0.962
    mAP@0.5:0.95: 0.941

- **Oxygen Tank**
  - **Precision:** 1.000
  - **Recall:** 0.920
  - **mAP@0.5:** 0.985
  - **mAP@0.5:0.95:** 0.946

These results confirm that the model performs consistently well across all object types, with particularly high precision and reliable recall rates.

3. Challenges and Solutions

The Challenge

One of the main challenges we encountered was class imbalance. Toolkit instances were significantly fewer compared to other objects. Additionally, we faced difficulties in maintaining consistent detection under varying lighting conditions and object occlusions.

How We Solved It

To address these issues, we introduced advanced augmentations like CLAHE and grayscale transformations. We also ensured that underrepresented classes were given more focus during annotation. Furthermore, label smoothing and mosaic augmentation helped the model generalize better, even with limited samples for certain classes. Early stopping and regular validation checks helped prevent overfitting.

4. Conclusion and Future Work

Summary of Achievements

We successfully developed a real-time object detection model capable of identifying space station safety gear with high precision and recall. The model shows strong generalization across different scenes and performs well on challenging cases such as partial visibility or variable lighting.

Future Improvements

While the current version of **Eleven11** delivers strong results in detecting essential safety equipment, there are several directions we aim to pursue to improve its functionality, reliability, and deployment-readiness:

- **Web and Mobile Deployment**
  Package the trained model into a fully functional web and mobile interface, allowing users to upload images or use their device camera for live safety gear detection.
- **3D Spatial Mapping of Objects**
  Integrate the model with a 3D map of the space station or industrial environment, allowing detected objects to be visualized in their actual spatial context.
- **Real-Time Missing Equipment Alerts**
  Implement a system that checks for the presence of required safety gear and alerts users if any item is missing or misplaced in the environment.
- **Continuous Learning via Falcon**
  Use Falcon to monitor performance in real deployments, collect real-world feedback, and automate retraining of the model as new object types or visual conditions emerge.
- **Expanded Object Categories**
  Add new object classes beyond the current three (oxygen tank, toolbox, fire extinguisher), such as safety helmets, gas detectors, first aid kits, etc.
- **Multi-language Accessibility**
  Add support for multilingual UI labels and voice descriptions to make the system accessible to a global user base, especially in international missions.
- **Audio-Based Alerts and Instructions**
  Integrate audio output to guide users in locating missing or low-confidence detections for accessibility and hands-free use.
- **Integration with Incident Reporting Tools**
  Automatically log detections or missing gear with timestamps and image evidence to link with safety reporting platforms for compliance and auditing.