Problem Statement and Team Details

Problem Statement: Train a YOLOv8-based object detection model to accurately identify key space station objects (Toolbox, Oxygen Tank, Fire Extinguisher) using synthetic data generated via Duality AI's Falcon digital twin platform.

Team Name: CODETECH

Team Leader Name: Shubham Kumar Jha

Institute Name: Central University Of Jammu

Theme Name: AI for Space Safety

Team Leader Email ID: sh23becse50@cujammu.ac.in

HACK WITH INDIA

Problem and Solution

Describe your problem statement and how you are solving it.

Problem: Astronauts must quickly locate critical items like fire extinguishers, oxygen tanks, and toolboxes in emergencies. Real-world training data is scarce in space missions.

Solution: Our solution uses Falcon's synthetic digital twin data to train a YOLOv8-based object detection model. This ensures high performance even in challenging space station conditions such as low light and occlusion.

Advanced Image Filtration Pipeline:

i. Applied ***CLAHE*** (Contrast Limited Adaptive Histogram Equalization) to enhance visibility under low-light conditions
ii. Introduced ***random Gamma correction*** for brightness simulation
iii. Added ***Gaussian noise to*** mimic sensor-level distortions
iv. Converted images to **LAB** and back to **RGB** to enhance contrast fidelity
v. Used a ***probability-based preprocessing switch*** to diversify training batches

Methodology & Implementation

Understanding the Dataset **:** We began by exploring Falcon's synthetic space station dataset. It had images simulating different lighting conditions, occlusions, and camera angles — making it ideal for training a model that needs to perform reliably in a real space environment.

Cleaning & Enhancing Images **:** Instead of using only traditional augmentations, we designed a custom image preprocessing system. This helped us create a more realistic and diverse training dataset.

Choosing and Configuring the Model : We selected YOLOv8m as our object detection model because of its strong performance and flexibility. We enabled segmentation support and fine-tuned its training settings. We used only ***30 epochs***, but with smart tuning, that was enough.

Applying Augmentations : To further improve performance, we enabled strong augmentations like :**Mosaic** (image mixing), **MixUp** (blending two images),**Copy-Paste** (object-level mixing) These techniques taught the model to handle unusual scenarios like overlapping objects or unusual layouts.

Training & Evaluation : We trained over ***50 different models***, adjusting hyperparameters like **learning rate**, optimizer (**AdamW),** and **momentum**. After each run, we compared models

Final Selection : From all models, **Train40** emerged as the best. It reached **95.8** accuracy . — giving us a powerful and deployable solution.

Technology Used

🧠 ***Model Architecture*** : YOLOv8m (Ultralytics) – We used the medium-sized variant with segmentation support for balanced speed and accuracy.

🏞️ ***Image Processing*** : • **OpenCV** – For applying CLAHE (contrast enhancement), gamma correction, noise simulation, and LAB color conversion.
• **NumPy** – Used extensively for matrix operations and image transformation logic.

🔄 ***Data Loading & Augmentation*** : • **Custom YOLOv8 Dataset Builder** – We integrated our preprocessing into YOLO's dataloader for on-the-fly training transformations.
• **Augmentations** : Mosaic, MixUp, Copy-Paste, Perspective, HSV shift, flipping, scaling, shearing.

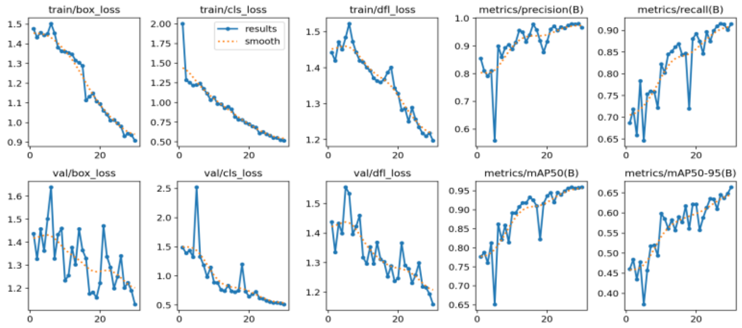
📈 ***Model Training & Evaluation :*** • **Python** – Our entire training pipeline was written in Python.
• **Argparse** – Used to pass configurable hyperparameters during training.
• **Google Colab** – Used for initial training experiments and logging.

📦 ***Environment & Dependencies*** : • ***Ultralytics Library*** – Core YOLOv8 functionalities
• **Conda/virtualenv** – Managed environments for reproducibility

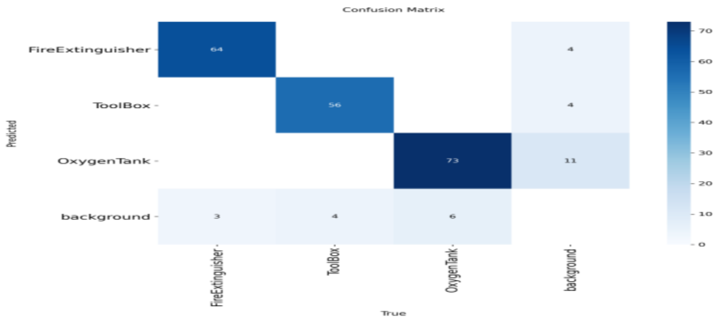
🧰 ***GitHub- Repo :*** [Codetech-BUILDWITHINDIA2.0]

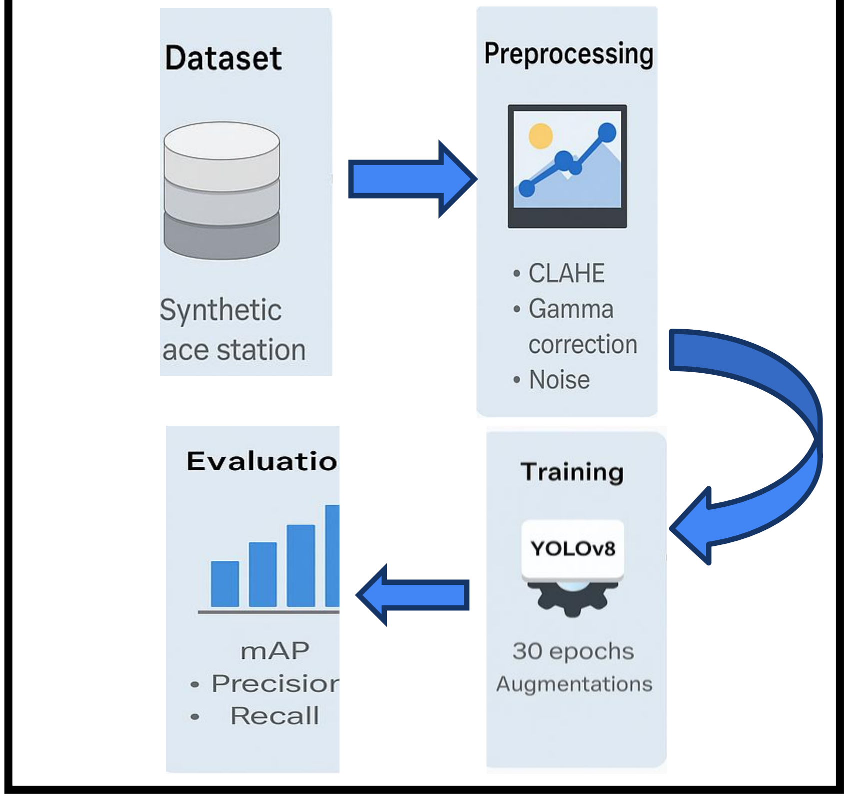
Flowchart & Supporting Images

🏆 Best Performance



Confusion Matrix





Feasibility and Market Use

🌍 ***Real-World Feasibility*** : Our solution is designed to work directly within synthetic or space-like environments. Since the model is trained on Falcon’s high-fidelity digital twin data, it doesn’t rely on real-world image collection — making it practical for space missions where data is hard to get.

It also requires only lightweight hardware (YOLOv8m), making it ideal for edge deployment on space station onboard systems or robotic units.

🚀 ***Market Applications:*** • **Space Stations** – Real-time detection of critical tools in emergencies (e.g., fire extinguishers, oxygen tanks).
• **Astronaut AR Assist** – Overlay detection in HUD displays for astronauts wearing AR visors.
• **Defense/Surveillance** – Monitoring equipment safety in submarines, bunkers, or aircraft where lighting and accessibility are limited.
• **Industrial Use** – Can be retrained for detecting machinery or safety gear in factories.

🛠️ ***Model Update Plan:*** Falcon can be used to simulate updated space environments

Conclusion

By using Falcon’s digital twin data and combining it with a carefully tuned YOLOv8 pipeline, we created a model that doesn’t just perform well — it adapts. Our custom image preprocessing and smart augmentation helped simulate real-world scenarios like lighting variations and occlusion.
We achieved:

- ✅ 95.8% training accuracy
- ✅ Robust generalization with just 30 epochs

💡 In a world where data collection is difficult, Our Model proves that synthetic data, when used smartly, can create powerful, practical AI solutions — even for space.
Successfully trained an object detection model for a simulated space station using synthetic data

🔮 What’s Next?

- Develop a deployable app for astronauts or engineers
- Add multi-angle camera support for enhanced detection
- Build a self-updating pipeline using Falcon for future retraining