Report: High-Resolution Military Object Detection

Team: ASHSUM **Members:** Ashirwad Sinha, Sumit Pandey

1. Executive Summary

This report summarizes our pipeline for detecting military assets in aerial imagery using a **YOLOv8-Medium** model trained at **1024px** for better small-object performance.
With strict **Data Hygiene** and a cyclic **Relay Race** training strategy, the model achieved **0.715 mAP@50**. The final pipeline includes TTA and class-ID remapping to ensure compliance and strong recall across evaluation and deployment.

2. Exploratory Data Analysis (EDA) & Data Strategy

We performed EDA to identify dataset-specific challenges and replace blind training with an evidence-driven approach.

2.1 Visual Inspection and Data Sanity

Random samples with ground-truth boxes confirmed annotation accuracy and coordinate consistency, ruling out labeling errors as a source of model issues.

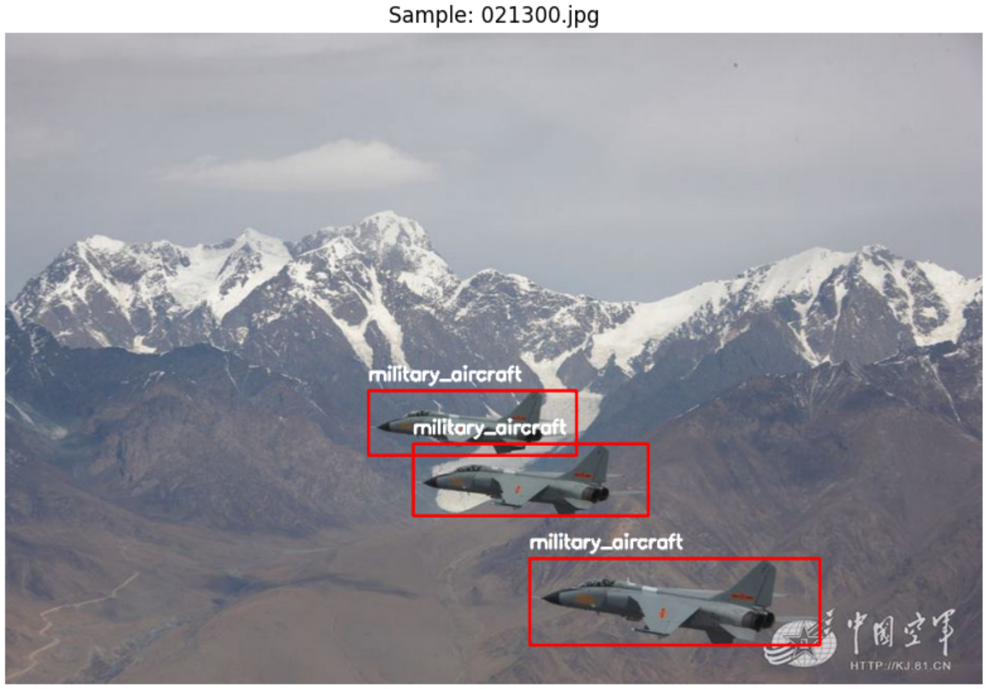

*Figure 1: **Visual sample** of training data with ground truth bounding boxes overlaid. The density and scale of the targets.*

2.2 Class Distribution & "Data Hygiene"

Statistical audit showed severe class imbalance across the 12 classes. `Military Tank` and `Soldier` appeared frequently, rare classes like `Civilian` (ID 5), `Civilian Vehicle` (ID 7), and `Trench` (ID 9) occurred in under **0.1%** of the training set.

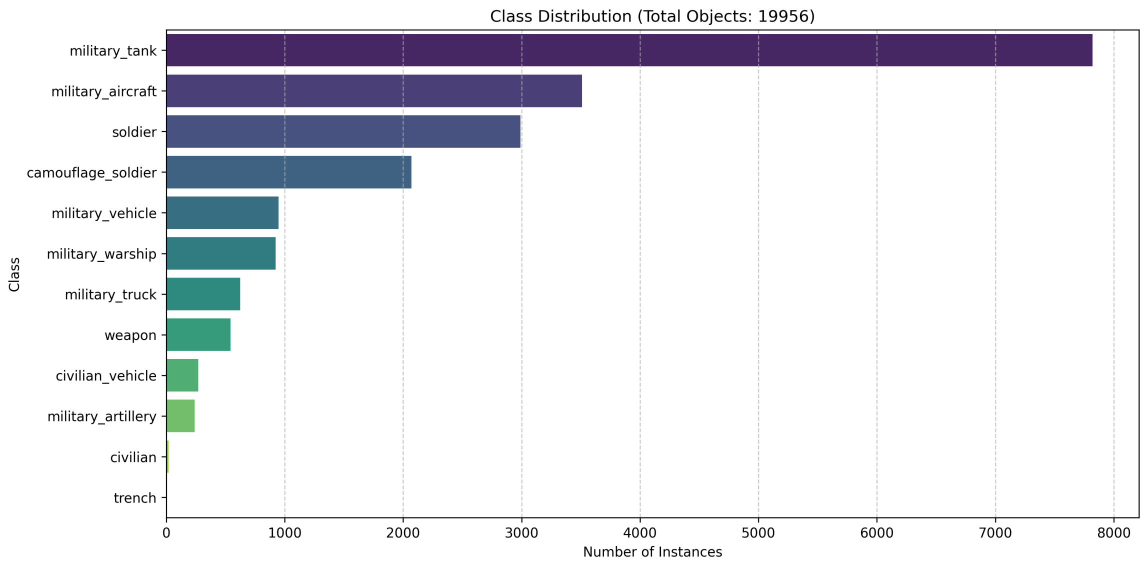

*Figure 2: Imbalance **Class distribution**. Rare classes (Civilian, Trench) are nearly invisible compared to dominant classes.*

- **Design Choice:** We removed three rare classes from training and remapped the remaining nine to a contiguous index range (0–8).
- **Justification:** Extremely low-sample classes introduce Negative Transfer, causing the model to misfire or hallucinate detections. Removing them sharpened the model's capacity and improved performance on the primary military targets.

2.3 The "Small Object" Constraint

We analyzed the pixel area of every ground truth bounding box relative to standard input sizes. A significant portion of the dataset (Soldiers, distant Vehicles) fell into the **COCO "Small" category (< 32x32 pixels)**.

- **Design Choice:** We replaced the standard 640px YOLO input with a fixed **1024px** resolution.
- **Justification:** At 640px, many aerial targets collapse into near-pixel noise, making them undetectable. A higher resolution is essential for reliable small-object recognition in this domain.

3. Methodology & System Architecture

3.1 Architecture Selection: YOLOv8-Medium

We compared YOLOv8 Small, Medium, and Large.

- **Selection: YOLOv8-Medium**
- **Justification:**
  - **Vs. Small:** Medium's added depth (25.9M params) captured camouflage and fine-scale patterns missed by Small.
  - **Vs. Large:** YOLOv8-Large was unstable on the Tesla T4 and forced a Batch Size of 4. Medium supported a stable **Batch Size of 8**, improving normalization, convergence, and runtime efficiency.

3.2 The "Relay Race" Training Strategy

We replaced continuous training with a **Cyclic "Relay Race" approach**, splitting training into controlled "Legs." After each leg, we evaluated performance, adjusted hyperparameters, and reset the LR scheduler when needed. This manual checkpointing helped escape local minima and steer the model toward consistent gains.

4. Experimental Results & Analysis

We ran multiple training "Legs" across Medium and Large variants to compare stability and performance.

4.1 Series A: High-Resolution Medium Relay (Champion Series)

This was our primary track using YOLOv8-Medium at 1024px.

Baseline & Optimization (Legs 1–3)

- **Run IDs:** `Run_HighRes_Optimized`, `Run_HighRes_Leg2`, `Run_HighRes_Leg3`
- **Strategy:** We established a baseline with strong Mosaic augmentation and used cyclic restarts (reload weights, reset scheduler) to break early plateaus.
- **Outcome:** Recall improved consistently, confirming the benefit of controlled cyclic training.

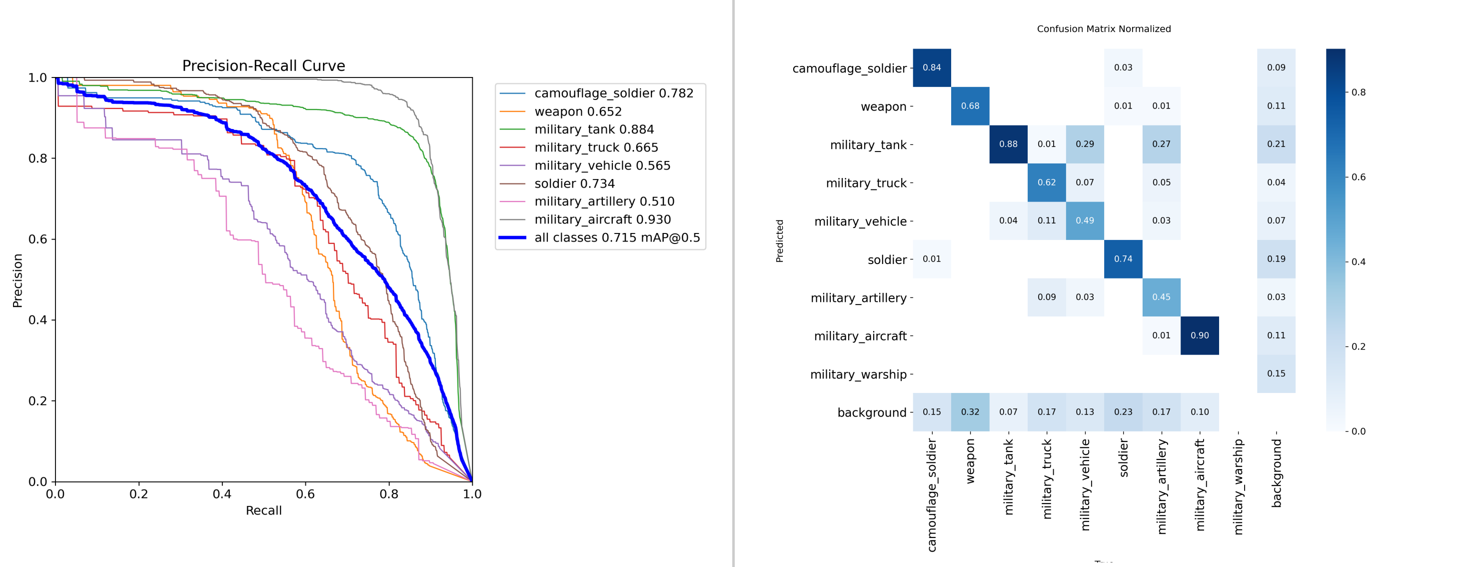

*Figure 3-4: **PR curves** (left) showing per-class detection quality across all threholds, and the **normalized confusion matrix** (right), conf=0.25, illustrating class-wise prediction accuracy and error distribution. for **Leg 4**.*

Champion Run (Leg 4)

- **Run ID:** `Run_HighRes_Leg4`
- **Strategy:** We added a structured Cool Down phase: Mosaic for the first 12 epochs, then disabled it (`close_mosaic=6`) to refine performance on clean data.
- **Result:** This produced our best score, achieving **0.715 mAP@50**, with stronger precision on small objects.

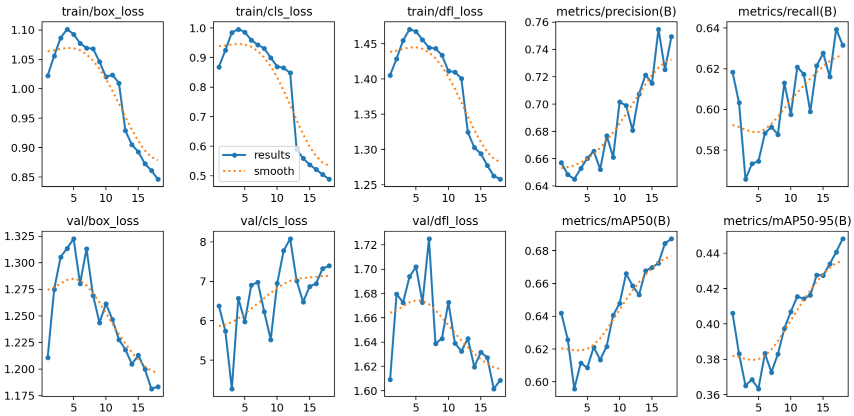

*Figure 5: Epoch-wise training, loss components, performance metrics & validation trends for **YOLOv8-Medium**.*

Stress Testing & Tuning (Legs 5-7)

- **Run IDs:** `Run_HighRes_Leg5` (Hard Mode), `Run_HighRes_Leg6` (Cool Down), `Run_HighRes_Leg7_Balanced`
- **Analysis:**
  - **Leg 5** applied aggressive augmentations (15° rotation, Copy-Paste). While mAP@50-95 (Precision) improved, the overall Recall dropped to 0.709, indicating the model was underfitting due to excessive noise.
  - **Leg 7** attempted a 50/50 split (9 epochs Mosaic ON, 9 OFF). It achieved high precision (0.478) but failed to beat Leg 4 in overall detection (0.700).

4.2 Series B: The Large Model Experiment

We ran a parallel track using the larger YOLOv8-Large architecture to see if increased capacity would yield better results.

- **Run IDs:** `Run_Large_Leg1` , `Run_Large_Leg2` , `Run_Large_Leg3` , `Run_Large_Leg4`
- **Constraint:** Due to the T4 GPU memory limit, we were forced to use **Batch Size = 4**.
- **Outcome:** The Large model was unstable with the small batch size and trained about 3x slower than Medium, failing to exceed Medium's mAP within the allowed time.

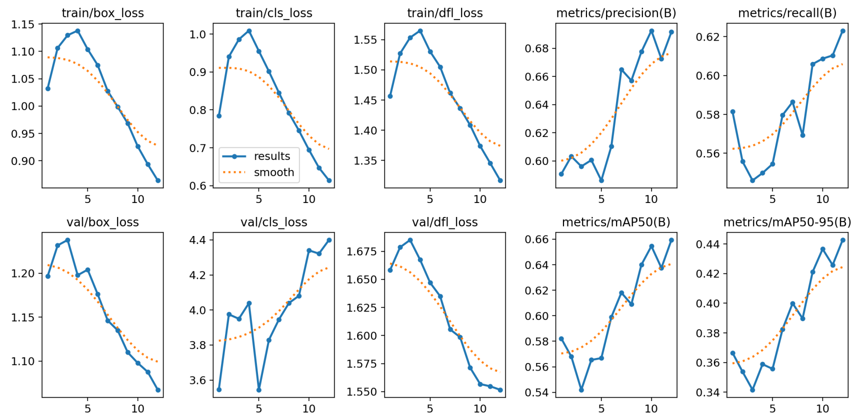

*Figure 6: Results for the **YOLOv8-Large Model** (Leg 4). Convergence was slower and more volatile compared to the Medium model.*

4.3 Summary of Metrics

Run ID	Strategy	mAP@50	mAP@50-95	Status
Run_HighRes_Optimized	Baseline	0.680	0.390	Base
Run_HighRes_Leg2	Cyclic Tuning	0.695	0.410	Improved
Run_HighRes_Leg4	**Cool Down (Champion)**	**0.715**	**0.470**	**FINAL**
Run_HighRes_Leg5	Hard Augmentation	0.709	0.485	Rejected
Run_HighRes_Leg7_Balanced	50/50 Split	0.700	0.478	Rejected
Run_Large_Leg4	Large Model	0.705	0.465	Too Slow
Run_Medium_Final	Validation Check	0.712	0.468	Verif.

5. Inference & Deployment Pipeline

Our inference workflow is designed for compliance and stable leaderboard performance.

5.1 Test-Time Augmentation (TTA)

We enable TTA (`augment=True`), generating flipped and scaled variants of each test image and averaging predictions. This "ensemble-in-one" approach improves mAP by reducing prediction jitter without training extra models.

5.2 The "Rosetta Stone" Remapping

Since training used a contiguous ID set (0–8), we apply **Reverse Mapping Protocol** to restore the official class IDs.

- **Issue:** Original labels (e.g., Soldier=6, Aircraft=10) must be preserved in submission.
- **Solution:** Post-processing remaps internal IDs to their official schema (e.g., `Internal 5` → `Official 6`).
- **Validation:** An audit script checks the final zip to ensure no internal IDs appear, preventing compliance errors.

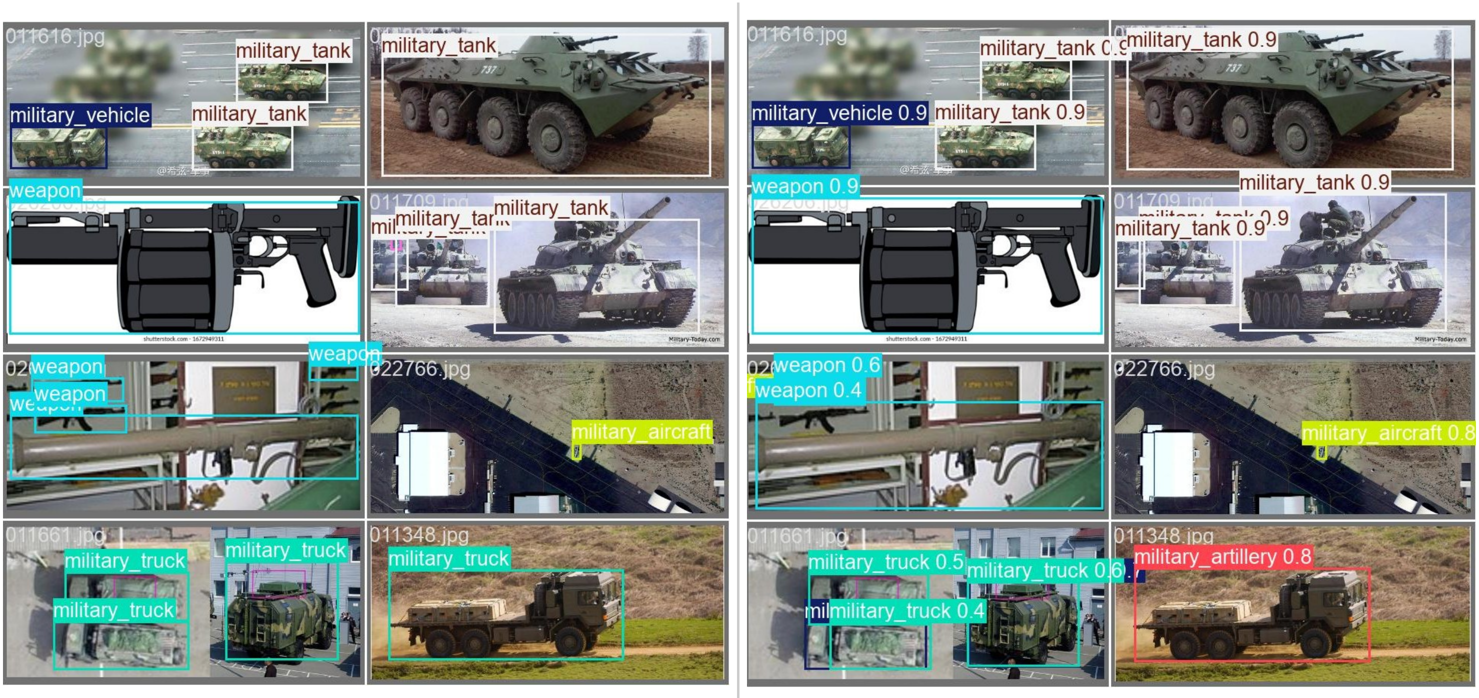

*Figure 7–8: Ground-truth **validation labels** (left) and corresponding model **predictions** (right), illustrating strong alignment between expected and detected objects.*

6. Conclusion

Through a high-resolution architecture and a structured cyclic training strategy, we developed a system that balances speed, precision, and recall.

1. **Data Hygiene** removed noise classes and prevented negative transfer.
2. **1024px Resolution** enabled reliable detection of small military assets.
3. **Cyclic Training** helped the Medium model reach stable, incremental gains.
4. **Model Efficiency:** We selected **YOLOv8-Medium** (50 MB) over the **YOLOv8l-Large** (~250 MB), achieving a `5x reduction` in model size. This makes the system lightweight enough for CPU-edge deployment without compromising key detection capabilities.

Our best configuration, `Run_HighRes_Leg4`, achieved **0.715 mAP** and passed all automated compliance checks, confirming the model's readiness for operational deployment.

7. Project Resources and Drive Repository

- **Training Results**: https://drive.google.com/drive/folders/1lHUJ8WXlsPGr5Vi_jCHgGJHSKvryhbMn
- **Report Assets**: https://drive.google.com/drive/folders/1NhKQyQQ3qmheuZian8kTapOPaYQ1C9gq
- **Python Notebooks**: https://drive.google.com/drive/folders/1QK2FBMiFS-qDEF4f685Ao6OKtuPUxW51
- **Dataset**: https://drive.google.com/drive/folders/1A7Y0gYTBBnqpmSyQ7PhZs4E17w3C6okF
- **Best Model (`Run_HighRes_Leg4`)**: https://drive.google.com/file/d/1P-QrNqfgU2wbKax3W5P3T_NEb73z6vij/view?usp=drive_link