

# Multi-Class Object Detection under Real-World Constraints

## 1. Introduction: -

Object detection in a real-world setting is a challenging computer vision task due to variations in terms of object size, lighting, background, and occlusions. This challenge worsens when the application considers military and surveillance objectives, where a stable detection is a critical component for downstream decisions being made.

The proposed solution responds to the challenge of multi-class object detection posed by a 12-class military object dataset, aiming at creating an end-to-end deployable system for detection. In addition to accuracy, robustness and complexity of computation, which correspond to realistic deployment requirements, serve as evaluation standards.

As the main objective is to construct and train a detector network that maintains high mAP@50 values while supporting speedy convergence and computationally efficient inference on commodity GPUs and CPU architectures.

## 2. Data Description and Pre-processing: -

### 2.1 Dataset Structure

The data set is in the standard YOLO directory structure as follows:

**Training set:** 10,000 images

**Validation set:** 2,941 images

**Test set:** 1,396 images (labels withheld)

Each image is annotated with bounding boxes from the YOLO format for 12 different classes of objects, ranging from military vehicles to personnel and weapons, as well as civilian-related objects. The dataset shows a large amount of class imbalance and also has a non-trivial number of images consisting of the background alone.

## 2.2 Pre-processing

Images are all resized to a 640×640 size while considering aspect ratio. The labels are checked for correctness and accuracy in relation to the image size. During the scanning process for the dataset, a few annotations included segmentation data along with the bounding boxes. Given the strict nature of the object detection task, only the bounding boxes were considered while the segmentation data remained unused.

## 2.3 Data Aug

In order to enhance the generalization performance under changing real-world conditions, the training process included the following augmentations:

- Mosaic augmentation • HSV color jitter • Random horizontal flipping
- Mix-up and Copy-Paste • Rand Augment • Random erasing

These augmentations were chosen specifically to address common failure cases related to scale variation, lighting changes, and partial occlusions in imagery relevant to the military.

## 3. Model Architecture and Training Strategy: -

### 3.1 Model Selection

A YOLOv8-Medium model (YOLOv8m), known for its good compromise between detection accuracy and speed, has been chosen for this project. Though more advanced architectures exist, they require more processing power, so a YOLOv8m model is more appropriate for mid-range GPU-CPU computers. The weights were initialized with weights that are pre-trained on COCO, which helped to speed up the convergence and generalize over the limited domain datasets.

### 3.2 Training Configuration

- Input resolution: **640 × 640** • Optimizer: **AdamW** • Initial learning rate: **0.001**
- Learning rate schedule: **Cosine decay** • Batch size: **16** • Epochs: **60**

- Mixed Precision Training (AMP): Enabled
- Hardware: **2 × Tesla T4 GPUs** • Distributed Data Parallel (DDP): Enabled

## 4. Validation Metrics and Results: -

### 4.1 Validation Results

- mAP@50:  $\approx 0.5326$
- mAP@50–95:  $\approx 0.3482$
- Precision:  $\approx 0.6706$
- Recall:  $\approx 0.4996$

### 4.2 Performance Analysis

The model demonstrates consistent improvement across epochs, with no evidence of overfitting at the observed stage. Precision-recall trends indicate improving object localization and class discrimination.

Lower mAP@50–95 values suggest that fine-grained localization remains challenging, particularly for small or partially occluded objects such as weapons and camouflage soldiers. This behaviour is consistent with the dataset’s visual complexity and class imbalance.

## 5. Inference Pipeline and Submission Format: -

After training completion, inference is performed on all test images using the best validation checkpoint. Predictions are exported in **YOLO TXT format**, with each detection line containing:

```
class_id x_center y_center width height confidence
```

Each prediction file strictly matches the corresponding test image basename. All output .txt files are packaged into a single ZIP archive to ensure compatibility with automated evaluation.

## 6. Inference/Test: -

### Configuration:

- Image Size: 640
- Confidence Threshold: 0.25
- IOU Threshold: 0.45
- Max Detections: 300

### Results:

- Total Images: 1396
- Images with Detections: 1250
- Total Detections: 2370
- Average Detections per Image: 1.70

### Detections per Class:

- military\_tank: 980 (41.4%)
- camouflage\_soldier: 393 (16.6%)
- soldier: 382 (16.1%)
- military\_aircraft: 243 (10.3%)
- military\_warship: 125 (5.3%)
- military\_truck: 85 (3.6%)
- military\_vehicle: 77 (3.2%)
- military\_artillery: 52 (2.2%)
- civilian\_vehicle: 23 (1.0%)
- weapon: 9 (0.4%)
- civilian: 1 (0.0%)

### Detection Summary:

Total Detections: 2370  
Images with Detections: 1250/1396  
Detection Rate: 89.5%