# RT-DETR Performance Report on WaRP-D

## Objective

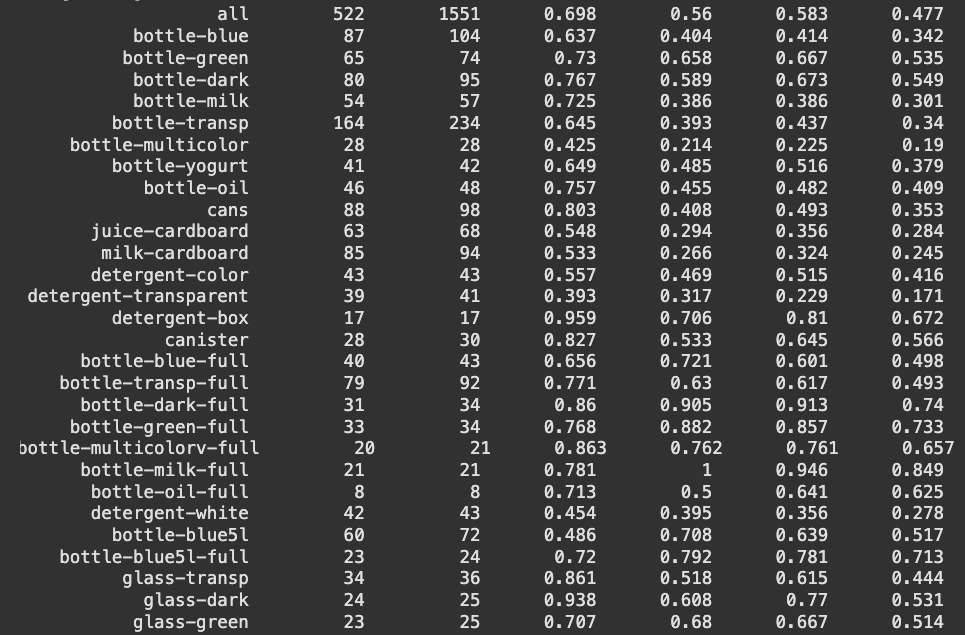
To analyze the detection performance of RT-DETR on the WaRP-D dataset using Ultralytics, with a focus on high-confidence (conf=0.6) validation for deployment-ready results.

## Experiment Settings

**Training configuration:**  
**Model:** RT-DETR-l, pretrained on COCO for strong starting weights.  
**Epochs:** 150 full passes through the data for thorough learning.  
**Batch size:** 16 images per step, balancing speed and stability.  
**Image size:** All images resized to 640×640 for consistent input.  
**Augmentations:**  
*Mosaic 0.5:* Combines 4 images for variety (50% of batches).  
*Mixup 0.1:* Blends 2 images (10% of batches).  
*HSV h=0.02, s=0.8, v=0.5:* Color/brightness changes for robustness.  
**Learning rate:** 0.001 for stable, gradual weight updates.  
**Label smoothing:** 0.1, to reduce overconfidence and handle label noise.  
**Early stopping:** Off (patience=0), always trains all epochs.  
**TTA:** Enabled at validation for more robust predictions.

Result snapshot:  
- mAP50: 0.583  
- mAP50–95: 0.477

Training Results (Best Model):



## 2. Validation Phase

**Validation Experiment :**  
results = model.val(  
 data='/kaggle/working/data.yaml',  
 imgsz=640,  
 conf=0.6, # Only keep predictions with 60%+ confidence  
 augment=True # Test Time Augmentation for extra robustness  
)

**Result snapshot:**- mAP50: **0.648**  
- mAP50–95: **0.548**  
  
At **conf=0.6,** only detections with >60% confidence are kept, giving results that focus on the model’s most certain predictions. This increases precision and is suitable for real-world deployment scenarios.

**Why Use conf=0.6?**  
Keeps only predictions with 60%+ confidence, reducing false positives and making results more reliable.

**Benefits:**  
Higher precision (fewer “junk” boxes), cleaner outputs.  
More trust for real-world use (e.g., recycling robots, quality control).  
mAP50/mAP50–95 reflect only the most certain detections.

**When to Use:**  
In production or deployment, where mistakes are costly and precision matters more than catching every possible object.

**Future Work:**  
To further improve:  
Try more/different augmentations  
Test other confidence levels (0.5, 0.7)  
Use TTA for extra robustness

### Validation Results:

## 

## Conclusion

RT-DETR-l delivers robust and high-precision results, especially at high confidence (conf=0.6). Always monitor both overall and per-class metrics to fully understand model performance.