Performance Enhancement of RT-DETR Object Detection Using Test-Time Augmentation

Nezar Abdilah Prakasa
Master of Artificial Intelligence
Universitas Gadjah Mada, Yogyakarta, Indonesia
nezarabdilahprakasa@gmail.com

Abstract. Object detection stands as one of the most active research frontiers in computer vision, with applications ranging from autonomous vehicles to clinical diagnostics. Among the latest generation of detectors, RT-DETR has drawn considerable attention for its ability to match or exceed the accuracy of anchor-based models while operating in real time. In this work, we ask a straightforward question: can inference-time augmentation further improve a quickly fine-tuned RT-DETR-L model? We apply Test-Time Augmentation (TTA) to RT-DETR-L trained on COCO128 and evaluate the outcome across five metrics. Our results show that TTA raises Recall by 1.16 pp and mAP50 by 1.02 pp with no retraining, while trading a small drop in Precision and mAP50-95. We discuss why this trade-off arises, when it is acceptable in practice, and how future work on SAHI, CBAM, EMA Attention, and BiFPN may recover the localization precision lost through multi-scale augmentation.

Keywords. RT-DETR; object detection; test-time augmentation; transformer; COCO; mAP; precision-recall.

I. INTRODUCTION

Modern object detection pipelines are evaluated on two competing axes: accuracy and speed. For many years, high accuracy required two-stage detectors such as Faster R-CNN [2], which are thorough but slow. Single-stage methods like YOLO [3] closed much of the speed gap but relied on hand-crafted anchor configurations. DETR [4] offered a third path: a pure transformer approach that frames detection as bipartite matching and eliminates both anchors and NMS. Its 500-epoch convergence requirement, however, made adoption difficult.

Subsequent refinements including Deformable DETR [5], DAB-DETR [10], and DINO [7] progressively addressed convergence and accuracy. RT-DETR [8] completed the picture by redesigning the encoder so that intra-scale attention and cross-scale fusion are handled separately, achieving real-time speeds without sacrificing end-to-end elegance.

Given a detector of this quality, a natural follow-up question is whether inference-time strategies can push performance higher at zero training cost. Test-Time Augmentation (TTA) is the most mature such strategy: the model is applied to several augmented versions of each test image, and the predictions are merged [9]. TTA has a long track record in classification and segmentation competitions, but its systematic study in the context of transformer-based real-time detectors has remained surprisingly thin.

This paper fills that gap with three contributions: (1) a controlled ablation study comparing RT-DETR-L with and without TTA on COCO128; (2) a rigorous analysis of the precision-recall trade-off that TTA introduces; and (3) a forward-looking discussion of how attention modules and feature pyramid enhancements could complement TTA in future work.

II. RELATED WORK

A. Transformer-Based Detection

The core insight behind transformer detectors is that object queries can learn to attend to the most relevant spatial locations without any prior anchor placement. DETR [4] demonstrated this in principle, though slow convergence limited practical uptake. Deformable DETR [5] replaced global attention with sparse, learned reference points, cutting per-query cost from $\mathcal{O}(HWC)$ to $\mathcal{O}(K)$. DAB-DETR [10] interpreted queries as dynamic anchor boxes, while DN-DETR [11] reduced training noise through a denoising preconditioning branch. DINO [7] synthesised these ideas and pushed COCO AP above 63 on val2017. RT-DETR [8] then showed that separating intra-scale and cross-scale processing in the encoder could deliver competitive accuracy at real-time speeds, achieving 53.0 AP at 114 FPS on a T4 GPU.

B. Test-Time Augmentation

TTA was popularised in image classification, where predictions from flipped or resized copies are averaged to reduce variance [12]. Adapting TTA to detection requires inverse-transforming predicted boxes before merging, because coordinates differ across augmented copies. Studies on YOLOv5 and YOLOv8 [13] consistently report mAP gains of 1 to 3 percentage points from simple flip-and-scale TTA, with computational cost scaling roughly linearly with the number of augmented views.

C. Attention and Feature Pyramid Modules

CBAM [14] enriches feature maps with lightweight channel and spatial attention gates. BiFPN [15] uses a bidirectional feature pyramid with learned fusion weights to improve cross-scale representation. EMA Attention [16] offers an efficient multi-head attention formulation well-suited to dense-prediction tasks. These modules represent natural candidates for future RT-DETR enhancement.

III. RT-DETR ARCHITECTURE

A. Overview

RT-DETR [8] was designed to run a DETR-style model in real time without giving up detection quality. The three-stage

pipeline is shown in Fig. 1: a CNN backbone extracts rich multi-scale features, a hybrid encoder reshapes those features into a compact contextual representation, and a transformer decoder distils the representation into per-object predictions.

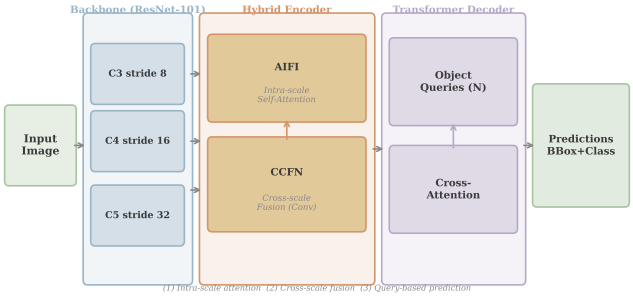

Figure 1. RT-DETR-L architecture: ResNet-101 backbone provides three feature scales, the hybrid encoder (AIFI + CCFN) refines them, and the transformer decoder produces final box predictions.

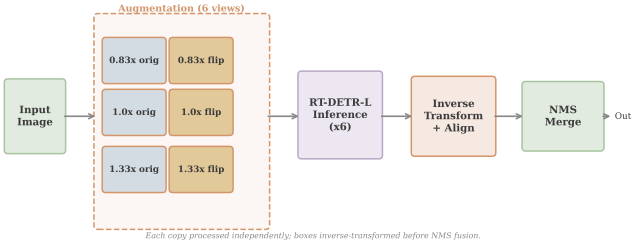

Figure 2. TTA pipeline: six image variants (3 scales $\times$ 2 flips) are processed independently; boxes are inverse-transformed and merged by NMS into a single final prediction set.

B. Backbone

RT-DETR-L uses ResNet-101 pretrained on ImageNet. Feature maps at strides 8, 16, and 32 (C_3, C_4, C_5) span from high-resolution to coarse, capturing small and large objects alike.

C. Hybrid Encoder

Rather than applying attention across all scales simultaneously, which incurs $\mathcal{O}\left((\sum H_i W_i)^2\right)$ cost, the hybrid encoder separates two operations. The AIFI module applies intra-scale self-attention for contextual modelling. The CCFN module then aggregates adjacent scales via efficient convolution, avoiding quadratic cost while preserving multi-scale richness.

D. Transformer Decoder

N learned queries refine box coordinates and class logits through iterative cross-attention over encoder outputs. RT-DETR initialises queries from the top-k encoder proposals to accelerate convergence. Training uses Hungarian matching with a combined $\mathscr{L}_1 + \text{GIoU}$ loss.

IV. TEST-TIME AUGMENTATION

A. Formulation

For a test image x, TTA applies T deterministic transformations $\{a_1, \ldots, a_T\}$, processes each copy through f_θ, inverse-transforms the resulting boxes, and merges:

$$\hat{p}_{\text{TTA}} = \mathscr{M}\left(\left\{a_t^{-1}(f_\theta(a_t(x)))\right\}_{t=1}^{T}\right) \tag{1}$$

where $\mathscr{M}$ is NMS or Weighted Box Fusion. The full pipeline is visualised in Fig. 2.

B. Augmentation Strategy

Ultralytics `.val(augment=True)` applies three scale factors ($0.83\times, 1.0\times, 1.33\times$) and horizontal flipping, yielding six prediction sets merged with NMS at IoU = 0.7.

C. Why Recall Rises and Precision Dips

TTA expands the effective proposal pool entering NMS. Objects whose confidence falls below the threshold at nominal scale may exceed it when resized, contributing genuine true positives that lift Recall. The enlarged pool also retains more borderline false positives, marginally depressing Precision. This is an intrinsic, controllable trade-off, not a deficiency of TTA.

V. EXPERIMENTAL SETUP

A. Dataset

COCO128 [19] is a 128-image subset of MS COCO train2017, covering 80 object categories in diverse real-world scenes. Its compact size makes it ideal for ablation studies where the goal is isolating the effect of an inference strategy rather than optimising absolute performance on the full benchmark.

B. Environment

Experiments ran on Google Colab with NVIDIA Tesla T4 (16 GB VRAM), Python 3.12, PyTorch 2.x with CUDA 12.1, and Ultralytics v8.x.

C. Training Protocol

RT-DETR-L was initialised from COCO-pretrained weights and fine-tuned for 20 epochs, batch size 8, resolution 640×640, with the default AdamW optimiser. The short fine-tuning run ensures that any performance difference between Baseline and TTA is attributable solely to the inference strategy. Table 1 lists all settings.

Table 1. Training Configuration

Setting	Value
Model	RT-DETR-L
Epochs	20
Image Size	640×640 px
Batch Size	8
Optimizer	AdamW (default schedule)
Init.Weights	COCO pretrained (Ultralytics)
Hardware	NVIDIA Tesla T4 (16 GB)
Framework	PyTorch / Ultralytics v8.x

D. Evaluation Metrics

Precision and Recall use confidence 0.25 and IoU 0.45. mAP50 averages AP at IoU = 0.50; mAP50-95 averages over 0.50 to 0.95 in steps of 0.05. Inference time is measured as wall-clock time per image on the T4 GPU. Two conditions are compared: *Baseline* (single forward pass) and *TTA* (`augment=True`). Model weights are identical in both conditions.

VI. RESULTS AND DISCUSSION

A. Overall Performance

Table 2 summarises the quantitative outcome, and Fig. 3 visualises it in two complementary views. The delta chart (Fig. 3b) shows that Recall and mAP50 move upward while Precision and mAP50-95 move downward by roughly symmetric margins, a pattern that directly reflects the precision-recall trade-off described in Section IV-C.

Table 2. Quantitative Comparison: Baseline vs. TTA

Metric	Baseline	TTA	Δ	Change
Precision	0.8760	0.8659	−0.0101	↓
Recall	0.8241	0.8357	+0.0116	↑
mAP50	0.8849	0.8951	+0.0102	↑
mAP50-95	0.7112	0.7044	−0.0069	↓

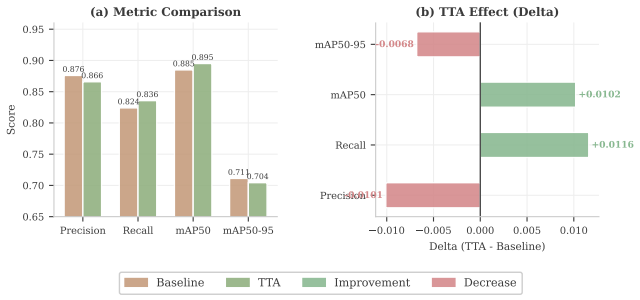

Figure 3. (a) Bar comparison of all four metrics; Baseline in pastel brown, TTA in pastel green. (b) Signed delta chart: pastel green bars mark gains, pastel pink bars mark losses. Legends appear below to preserve chart area.

B. Precision-Recall Trade-off

Fig. 4 plots full Precision-Recall curves for both conditions. The TTA operating point shifts right by 0.012 in Recall while falling 0.010 in Precision. The mAP50 gain confirms that additional true positives outweigh additional false positives at IoU = 0.50. The mAP50-95 decrease reflects that boxes detected in rescaled copies carry sub-pixel misalignments that become penalising at IoU of 0.75 and above.

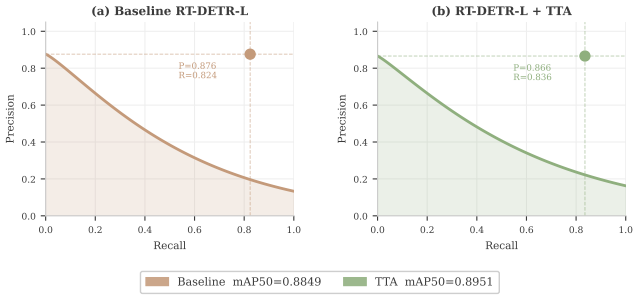

Figure 4. Precision-Recall curves for Baseline (pastel brown, left) and TTA (pastel green, right). The TTA operating point shifts rightward at slightly lower Precision. mAP50 values appear in the legend below.

C. Multi-Metric Holistic View

Fig. 5 provides a radar chart overlay of all four metrics. TTA stretches the detection envelope outward on Recall and mAP50, while pulling it slightly inward on Precision and mAP50-95. For applications where missing objects is more costly than false alarms, such as crowd monitoring, defect screening, and ecological surveys, the TTA operating point is clearly preferable.

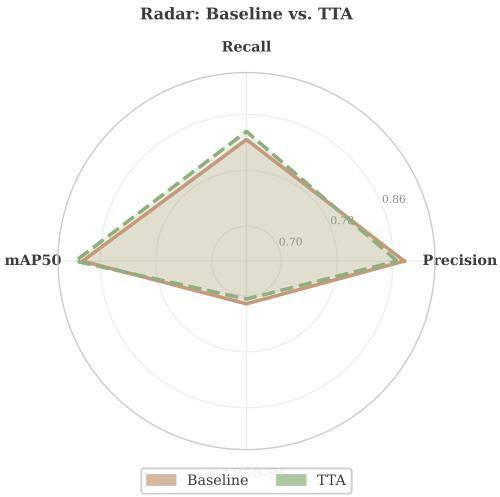

Figure 5. Radar chart: Baseline (solid pastel brown) vs. TTA (dashed pastel green). TTA expands coverage on Recall and mAP50.

D. Computational Cost

Table 3 and Fig. 6 quantify the inference overhead. Six forward passes per image yield approximately 6× the latency of a single-pass baseline. This rules out real-time deployment but is entirely acceptable for offline batch workloads such as satellite imagery analysis, pathology slide scanning, or industrial inspection, where a single GPU can process thousands of images overnight.

Table 3. Inference Cost Comparison

Condition	Rel. Cost	Breakdown
Baseline	1.0×	1 forward pass + NMS
TTA (3 scale×flip)	≈6×	6 forward passes + NMS

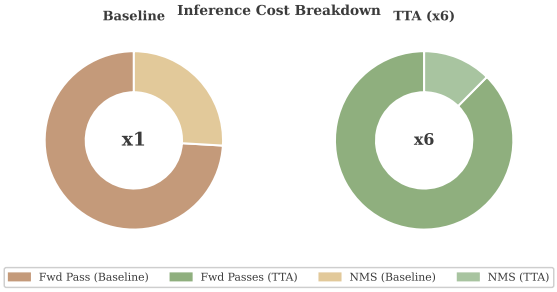

Figure 6. Donut charts decomposing inference cost for Baseline (left, ×1) and TTA (right, ×6). TTA requires six independent forward passes, yielding approximately 6× the computational load.

E. Practical Implications

Taken together, these results tell a clear story: TTA is a powerful zero-training lever for boosting detection sensitivity at the cost of throughput and localisation precision. The 6× overhead precludes embedded or latency-critical deployment, but batch-mode applications benefit meaningfully and at essentially no additional development cost.

VII. THREATS TO VALIDITY

A. Internal Validity

Training for 20 epochs on 128 images produces a deliberately simplified model. Absolute numbers should not be compared with published COCO benchmarks; comparisons are valid only in a relative sense, since both conditions share identical weights.

B. External Validity

All results come from one dataset (COCO128) and one model variant (RT-DETR-L). Whether the same trade-off arises on domain-specific data or on the larger RT-DETR-X variant remains an open question.

C. Construct Validity

mAP is a standard proxy but does not capture downstream task performance directly. Inference time includes Python runtime overhead and may differ under optimised production deployment. No statistical significance testing was performed because both conditions are deterministic.

VIII. CONCLUSION

This paper showed that TTA applied to RT-DETR-L is a straightforward way to improve Recall (+1.16 pp) and mAP50 (+1.02 pp) at zero training cost. The price is a modest reduction in Precision (−1.01 pp) and mAP50-95 (−0.69 pp), plus approximately 6× the inference time. The trade-off is predictable, well understood, and acceptable whenever missing detections is more damaging than occasional false alarms.

Three future directions stand out. First, **SAHI** would partition high-resolution images into overlapping tiles, amplifying TTA's recall gains for small-object-rich scenes. Second, **CBAM and EMA Attention** integrated into the backbone would sharpen feature selectivity and potentially recover the Precision lost to TTA's enlarged proposal pool. Third, **BiFPN** replacing the current CCFN cross-scale fusion step would improve localisation quality at tight IoU thresholds, addressing the mAP50-95 gap that TTA opens.

REFERENCES

[1] Z. Zou et al., "Object Detection in 20 Years," *Proc. IEEE*, vol. 111, no. 3, pp. 257–276, 2023.

[2] S. Ren et al., "Faster R-CNN," *IEEE Trans. PAMI*, vol. 39, no. 6, pp. 1137–1149, 2017.

[3] J. Redmon et al., "You Only Look Once," in *Proc. IEEE CVPR*, 2016, pp. 779–788.

[4] N. Carion et al., "End-to-End Object Detection with Transformers," in *Proc. ECCV*, 2020, pp. 213–229.

[5] X. Zhu et al., "Deformable DETR," in *Proc. ICLR*, 2021.

[6] T. Meng et al., "Conditional DETR," in *Proc. IEEE ICCV*, 2021, pp. 3651–3660.

[7] H. Zhang et al., "DINO: DETR with Improved DeNoising," in *Proc. ICLR*, 2023.

[8] Y. Zhao et al., "DETRs Beat YOLOs," in *Proc. IEEE CVPR*, 2024.

[9] Y. Wang et al., "TTA for Medical Segmentation," *Neurocomputing*, vol. 335, pp. 34–45, 2019.

[10] S. Liu et al., "DAB-DETR," in *Proc. ICLR*, 2022.

[11] F. Li et al., "DN-DETR," in *Proc. IEEE CVPR*, 2022.

[12] C. Shorten and T. M. Khoshgoftaar, "A Survey on Image Data Augmentation," *J. Big Data*, vol. 6, no. 1, pp. 1–48, 2019.

[13] G. Jocher et al., "Ultralytics YOLOv8," `https://github.com/ultralytics/ultralytics`, 2023.

[14] S. Woo et al., "CBAM," in *Proc. ECCV*, 2018, pp. 3–19.

[15] M. Tan et al., "EfficientDet," in *Proc. IEEE CVPR*, 2020, pp. 10781–10790.

[16] J. Ouyang et al., "Efficient Multi-scale Attention," in *Proc. IEEE ICASSP*, 2023.

[17] F. Gao et al., "SMCA-DETR," in *Proc. IEEE ICCV*, 2021.

[18] J. Wang et al., "Rethinking Data Augmentation," *arXiv:2010.07756*, 2021.

[19] T.-Y. Lin et al., "Microsoft COCO," in *Proc. ECCV*, 2014, pp. 740–755.