Comparative Analysis of RT-DETR for Detecting Trees in Aerial Images

Carlos Joel Tavares da Silva - 241111309
University of Brasília

***Abstract*—Accurate detection of individual trees in urban areas is essential for sustainable urban forest management. This study evaluates RT-DETR, a real-time, end-to-end object detection model, using high-resolution aerial image of urban forests. We compare RT-DETR's performance against a benchmark of 21 state-of-the-art object detection methods, focusing on the AP-50 metric. Results show that RT-DETR outperforms the benchmark models, achieving superior accuracy and robustness, particularly in complex urban forest environments with overlapping canopies and varied tree densities. These findings highlight RT-DETR's potential. The project page: https://github.com/CJTS/cibernetica_trab1**

***Index Terms*—Deep Learning, Image Detection.**

I. Introduction

Urban forests are vital for ecological balance and livability, particularly in rapidly urbanizing cities. Detecting individual trees from aerial images poses unique challenges, including occlusions and varying tree densities. Deep learning models such as YOLO, Faster R-CNN, and RetinaNet have emerged as effective tools for such tasks.

The history of machine learning (ML) has been significantly shaped by advances in image analysis. From the early days of artificial neurons, such as the perceptron, to contemporary deep learning models, the ability to extract meaningful patterns from visual data has been a cornerstone of ML evolution. The emergence of Convolutional Neural Networks (CNNs) marked a pivotal point, revolutionizing object detection and image classification tasks.

YOLO [1], as an anchor-based one-stage detector, represents a culmination of decades of iterative improvements, offering real-time detection with remarkable speed. However, as ML models have matured, their gains in overall efficiency have started to plateau, especially in terms of precision and recall. The incremental benefits of newer models often come at the expense of higher computational demands, which can limit their accessibility and scalability in resource-constrained settings, e.g., real-time applications.

Recent research efforts have pivoted towards achieving comparable results with fewer resources. Lightweight architectures, model compression techniques, and hardware-efficient designs are increasingly being explored to balance performance with practicality. This shift reflects the growing demand for sustainable ML solutions, particularly in applications such as urban forestry, where resource efficiency is paramount.

The main goal of this article is to test RT-DETR [2], a real-time end-to-end object detector, that tested better than other models using the COCO Dataset [3] from 2017, on a tree detection benchmark and compare its results to check if it remains better and outperforms the models.

The rest of this article is composed of Method II, Results and Discussion III and Conclusions IV.

II. Method

A. Theory

YOLO [1] introduces an approach to object detection by framing it as a regression problem to predict bounding boxes and class probabilities directly from full images. This unified architecture processes the entire detection pipeline in a single neural network, enabling end-to-end optimization for detection performance. YOLOv1 was fast, achieving real-time processing at 45 frames per second. Although YOLO makes more localization errors, it reduces false positives, outperforming traditional methods like DPM and R-CNN when applied to diverse datasets such as the Picasso and People-Art datasets. A problem that arises from YOLO is that it requires Non-Maximum Suppression (NMS) for post-processing, which slows down the speed for inference and introduces hyperparameters that cause instability in both the speed and accuracy.

DETR (DEtection TRansformer) [4] redefines object detection as a direct set prediction problem, removing the need for traditional hand-designed components like NMS and anchor generation. It employs a transformer encoder-decoder architecture and a set-based global loss to enforce unique predictions via bipartite matching. Using a fixed set of learned object queries, DETR captures object relationships and global image context to output predictions in parallel. The model is simple, library-independent, and matches the performance of Faster R-CNN on the COCO dataset, while also excelling in panoptic segmentation with significant improvements over baselines.

RT-DETR (Real-Time DEtection TRansformer) [2] is a real-time, end-to-end object detector designed to balance speed and accuracy. It enhances DETR by employing an efficient hybrid encoder for fast multi-scale feature processing and a decoupled approach to intra-scale interaction and cross-scale fusion for improved speed. Additionally, RT-DETR introduces uncertainty-minimal query selection to deliver high-quality initial queries to the decoder, enhancing accuracy.

Figure **??** displays an overview of how RT-DETR works. It utilizes the features from the final three stages of the backbone as input to the encoder. The efficient hybrid encoder processes these multi-scale features into a unified sequence through the Attention-based Intra-scale Feature Interaction

(AIFI) for intra-scale processing and the CNN-based Cross-scale Feature Fusion (CCFF) for integrating information across scales. Subsequently, the uncertainty-minimal query selection mechanism identifies a fixed number of encoder features to initialize object queries for the decoder. The decoder, equipped with auxiliary prediction heads, then iteratively refines these queries to output the final object categories and bounding boxes.

B. Experiments

Diverse tree species were mapped using high-resolution RGB aerial images. The dataset includes 3382 annotated tree crowns. They are RGB high-resolution orthoimages of 5619 x 5946 pixels and a ground sample distance (GSD) of 10 cm. The images, collected in 2013 by the city hall of Campo Grande, was manually annotated using QGIS software.

For analysis, the orthoimages were divided into 220 non-overlapping patches of 512 x 512 pixels, representing an area of 51.20 x 51.20 meters (2621.44 m^2) per patch. Figure 2 denotes an example of a 512 x 512 pixels image and Figure 3 the same image with the border boxes of the trees.

The annotated dataset was originally used to benchmark 21 state-of-the-art deep-learning methods, including anchor-based (one, two, and multi-stage) and anchor-free detectors. The RT-DETR method, tested on the COCO dataset from 2017, demonstrated promising results, further motivating its evaluation in this specific urban forest application.

The experimental setup involved training the models over 72 epochs with a batch size of 8 and 4 workers for data loading. The dataset comprised 220 images using the COCO Dataset notation, which were divided into 70% for training and 30% for testing. The backbone used was ResNet-50, a widely adopted architecture known for its balance of efficiency and accuracy.

Figure 4 illustrates the workflow, where blue circles represent inputs, purple rectangles denote processes, and green diamonds indicate results. The process begins with an input treatment program, which takes the 220 images and splits them into 70% for training and 30% for testing. The images are formatted according to the COCO Dataset specification, including the necessary JSON annotations. Next, the RT-DETR model is fine-tuned using the training set, and the Average Precision at IoU 0.50 (AP-50) metric is calculated to evaluate the model's performance during testing.

III. Results and Discussion

During each epoch execution, detailed metrics are displayed to monitor the model's performance. These metrics include:

- Average Precision (AP): mean precision calculated across multiple IoU thresholds, typically ranging from 0.50 to 0.95. It indicates the model's ability to correctly predict bounding boxes and classify objects.
- Average Recall (AR): mean recall, which evaluates how many true objects were successfully detected by the model.

For each metric, the following parameters are considered:

TABLE I
EPOCH RESULT TABLE

	IoU	Area	MaxDets	Value
Average Precision (AP)	0.50:0.95	all	100	0,416
	0.50	all	100	0,765
	0.75	all	100	0,417
	0.50:0.95	small	100	0,263
	0.50:0.95	medium	100	0,468
	0.50:0.95	large	100	0,56
Average Recall (AR)	0.50:0.95	all	1	0,048
	0.50:0.95	all	10	0,343
	0.50:0.95	all	100	0,545
	0.50:0.95	small	100	0,431
	0.50:0.95	medium	100	0,577
	0.50:0.95	large	100	0,715

TABLE II
AP-50 RESULTS OVER THE 22 METHODS

Method	**AP-50**
YoloV3	0.591
Weight Standardization	0.631
RetinaNet	0.650
DetecoRS	0.651
Dynamic R-CNN	0.655
Deformable ConvNets v2	0.657
NAS-FPN	0.658
Faster R-CNN	0.660
SABL	0.661
VarifocalNet (1)	0.664
Generalized Focal Loss	0.677
Probabilistic Anchor Assignment	0.677
Mixed precision training	0.679
VarifocalNet (2)	0.683
Empirical Attention	0.690
Gradient Harmonized Single-stage Detector	0.691
ATSS	0.692
FoveaBox	0.692
CARAFE	0.697
Double Heads	0.699
FSAF	0.701
RT-DETR	0.756

- IoU (Intersection over Union): overlap ratio between the predicted and ground-truth bounding boxes.
- area: metrics are calculated for objects of different sizes: all, small, medium, large.
- maxDets (Maximum Detections): maximum number of detections considered during evaluation: 1, 10 or 100.

Table I shows an epoch output example. The first line output, for example, specifies that the AP is 0.416, calculated across IoU thresholds of 0.50 to 0.95, for all object sizes, considering up to 100 detections. The second line of each execution was the one used for calculating the metrics.

The results demonstrated in Table II show that RT-DETR outperforms in object detection tasks involving urban trees, as evidenced by its superior AP-50 score. This performance difference can be attributed to several key architectural advantages of RT-DETR over YOLO, the main method used as a comparative in Zhao *et al* (2023). Figure 5 shows a visual representation of table II.

The stabilization of loss and the use of metrics such as AP-50 are critical for evaluating and optimizing image detection

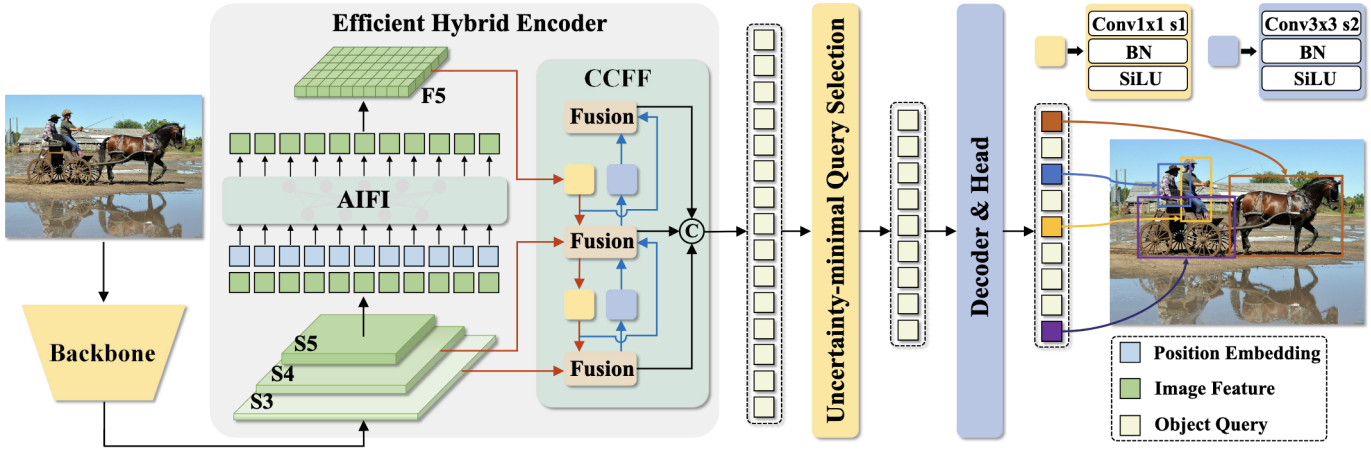

Fig. 1. Overview of RT-DETR. Source: [2]

Fig. 2. Example of 512x512 image: Source: [5]

Fig. 3. Example of 512x512 image with border box: Source: [5]

models. As a model progresses through epochs, the loss function ideally converges, indicating that the model is effectively learning to differentiate between objects and background. AP-50 serves as a key performance metric, providing insight into how well the model balances precision and recall at a specific IoU threshold. Stabilizing the loss (Figure 7) ensures that the model avoids overfitting while maintaining a consistent or improving AP-50 (Figure 6) metric indicates that the model's object detection capabilities are improving.

By analyzing Figure 8, which shows the AP-50 metric for RT-DETR, alongside Figure 9, presenting the top benchmark results, and Figure 10, depicting the mean performance of all 8 methods, valuable insights can be drawn. The top methods for

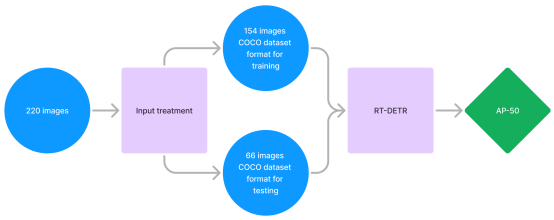

Fig. 4. Testing workflow: Source: [5]

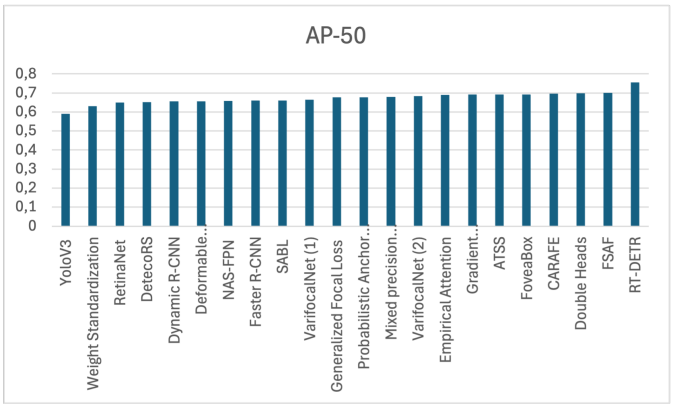

Fig. 5. Bar chart of AP-50 metrics of the 22 methods

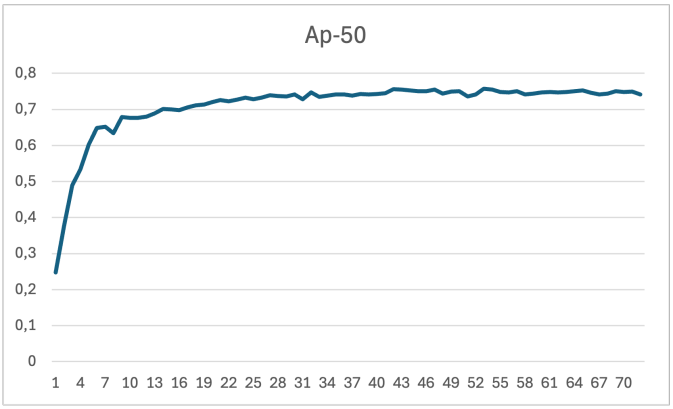

Fig. 6. AP-50 Line chart over 72 Epoch

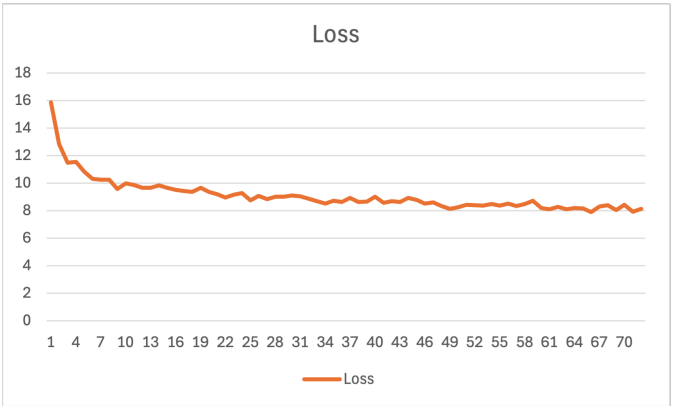

Fig. 7. Loss over 72 Epoch

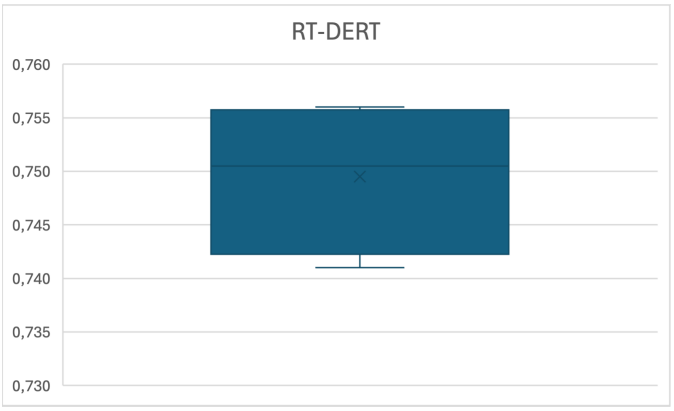

Fig. 8. Average AP-50

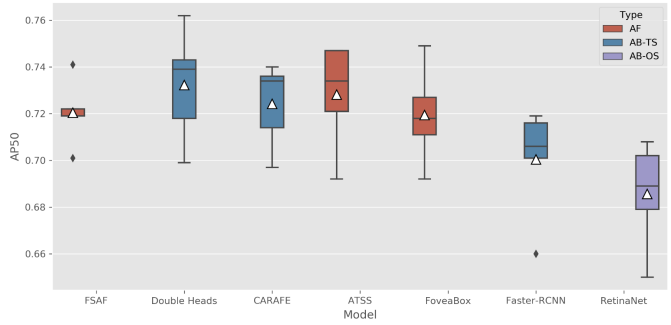

Fig. 9. Average AP-50. Source: [5]

object detection vary in design and focus, each offering unique strengths and weaknesses. Double Heads, ATSS, CARAFE, FSAF, and FoveaBox demonstrate incremental improvements in detection accuracy, with AP-50 values ranging from 0.719 to 0.732, indicating high precision at 50% IoU threshold. These methods often incorporate advanced feature extraction, enhanced anchor-free mechanisms, or refined region proposals to boost performance.

Double Heads and ATSS stand out for their ability to balance feature refinement with efficient region proposal generation, achieving an AP-50 of 0.732 and 0.728, respectively. Meanwhile, CARAFE and FSAF focus on enhancing feature aggregation and spatial reasoning, maintaining AP-50 scores around 0.720.

In contrast, Faster R-CNN and RetinaNet adopt a more traditional anchor-based approach, achieving lower scores of 0.700 and 0.686, respectively, as they rely more heavily on bounding box predictions and feature extraction from fixed anchors.

RT-DETR, however, stands out with an impressive AP-50 of 0.735. This method utilizes a transformer-based architecture, focusing on end-to-end detection without relying on region proposals or anchors. Its strong performance stems from its ability to leverage global contextual information, enhancing detection accuracy for complex scenes.

First, RT-DETR's transformer-based architecture allows it to effectively capture global context and object relationships

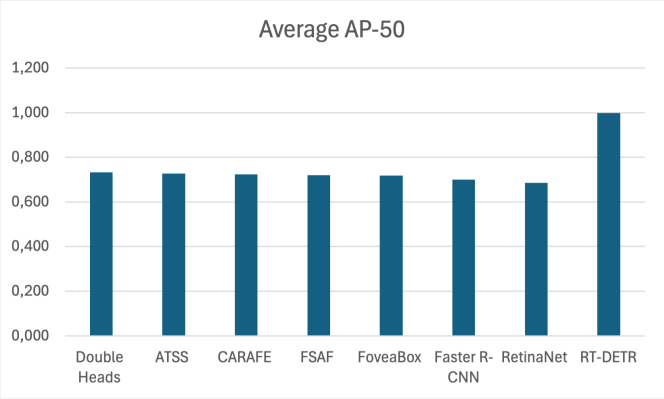

Fig. 10. Average AP-50. Source: [5]

within the image. This is particularly important for detecting trees in dense urban environments, where overlapping occlusions are common. Unlike YOLO, which relies on anchor-based mechanisms, RT-DETR employs an anchor-free approach that eliminates the limitations of predefined anchor boxes, enabling it to adapt more flexibly to objects of varying shapes and sizes.

Second, the hybrid encoder design in RT-DETR significantly enhances its ability to process multi-scale features. By decoupling intra-scale feature interactions (handled through Attention-based Intra-scale Feature Interaction) and cross-scale feature fusion (managed via CNN-based Cross-scale Feature Fusion), RT-DETR achieves a more efficient and accurate representation of features across different resolutions. This ensures that even small or partially occluded trees are accurately detected.

Additionally, the uncertainty-minimal query selection mechanism in RT-DETR contributes to its improved accuracy. By selecting high-quality encoder features as initial object queries, the decoder can focus on refining these queries with greater precision. This contrasts with YOLO's direct regression-based approach, which is more susceptible to errors in complex scenes.

The superior performance of RT-DETR aligns with recent literature that highlights the advantages of transformer-based models in object detection tasks. Studies [5] have demonstrated that anchor-free methods consistently outperform anchor-based detectors like YOLO in challenging scenarios, including those involving dense and diverse object distributions. Furthermore, the end-to-end design of RT-DETR simplifies the detection pipeline by removing the need for non-maximum suppression and other post-processing steps, which can introduce errors.

Overall, RT-DETR's design not only achieves better accuracy but also reflects a broader trend in object detection research: the shift towards models that leverage global attention mechanisms and adaptive feature processing. This makes RT-DETR particularly well-suited for applications in urban forestry, where the complexity of scenes demands robust and flexible detection capabilities.

IV. Conclusions

This study demonstrates RT-DETR's superiority over YOLO and other methods for detecting urban trees in aerial images. RT-DETR's ability to process multi-scale features and leverage a transformer-based architecture gives it a significant advantage in detecting objects in dense and occluded environments, as evidenced by its superior AP-50 metric.

Moreover, RT-DETR simplifies the detection pipeline by eliminating the need for NMS and other hand-crafted components, streamlining its application in real-world scenarios. Its anchor-free design also enables better adaptability to varying object shapes and sizes, further enhancing its robustness.

These findings suggest that RT-DETR is not only effective but also aligns with the growing trend of efficient and scalable object detection models. Its performance on a challenging dataset like urban forestry imagery highlights its potential for broader applications, including wildlife monitoring, urban planning, and precision agriculture.

Future research could explore hybrid architectures that combine the strengths of transformer-based and convolutional methods, as well as optimization techniques to further reduce resource consumption without sacrificing accuracy. Additionally, expanding the evaluation to include diverse datasets and environments would provide deeper insights into its generalization and performance.

In conclusion, RT-DETR sets a new benchmark for real-time object detection, particularly in scenarios requiring high precision and adaptability, reinforcing its status as a valuable tool in modern computer vision applications.

References

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016. [Online]. Available: https://arxiv.org/abs/1506.02640

[2] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, and J. Chen, "Detrs beat yolos on real-time object detection," 2023.

[3] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2015. [Online]. Available: https://arxiv.org/abs/1405.0312

[4] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," 2020. [Online]. Available: https://arxiv.org/abs/2005.12872

[5] P. Zamboni, J. M. Junior, J. d. A. Silva, G. T. Miyoshi, E. T. Matsubara, K. Nogueira, and W. N. Gonçalves, "Benchmarking anchor-based and anchor-free state-of-the-art deep learning methods for individual tree detection in rgb high-resolution images," *Remote Sensing*, vol. 13, no. 13, 2021. [Online]. Available: https://www.mdpi.com/2072-4292/13/13/2482