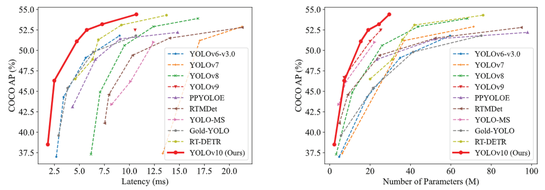
YOLOV10: REAL-TIME END-TO-END OBJECT DETECTION

AUTHORS: Ao Wang[1], Hui Chen[2*], Lihao Liu[1], Kai Chen[1], Zijia Lin[1], Jungong Han[3], Guiguang Ding[1*].

AFFILIATION: Tsinghua University (Departments: Software, Automation, BNRist).

1. INTRODUCTION

YOLO models are highly efficient for real-time object detection, yet they rely on Non-Maximum Suppression (NMS) for post-processing, which increases inference latency. Additionally, architectural redundancies limit their computational efficiency and performance. YOLOv10 addresses these challenges by eliminating NMS through consistent dual assignments and adopting an efficiency-driven design that reduces computational overhead while enhancing accuracy, setting a new benchmark for real-time object detection.



Comparisons with others in terms of latency-accuracy (left) and size-accuracy (right) trade-offs.

2. KEY INNOVATIONS

2.1. NMS-Free Training

YOLOv10 eliminates Non-Maximum Suppression, reducing inference latency with consistent dual assignments, ensuring efficient and accurate bounding box predictions.

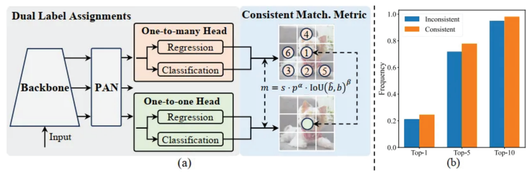
2.2. Efficiency-Accuracy Model Design

YOLOv10 optimizes performance with a lightweight classification head to reduce overhead, spatial-channel decoupled downsampling for efficient information retention, and rank-guided block design to minimize redundancy and improve efficiency.

2.3. Accuracy Enhancements

YOLOv10 improves accuracy through large-kernel convolutions, which expand receptive fields to capture more spatial context. Additionally, it incorporates Partial Self-Attention (PSA) to enhance global feature representation while maintaining computational efficiency, enabling more accurate object detection.

3. METHOD



3.1. NMS-Free Training

In previous YOLO models, Non-Maximum Suppression (NMS) was used in post-processing to eliminate redundant bounding boxes, ensuring that each object is represented by a single bounding box. However, NMS is computationally expensive and increases inference time, particularly when dealing with a large number of bounding boxes.

Backbone: Efficient Feature Extraction
The backbone of YOLOv10 is optimized for real-time performance through spatial-channel decoupled downsampling, which separates spatial and channel operations to reduce computational overhead while retaining critical feature information. Additionally, large-kernel convolutions are incorporated into deeper layers to expand receptive fields, allowing for improved spatial context capture and precise object localization.

Neck: Multi-Scale Feature Fusion
The neck of YOLOv10 incorporates a rank-guided block design that dynamically adjusts complexity across stages to minimize redundancy and optimize parameter usage. Additionally, it utilizes a path aggregation network to effectively fuse multi-scale features, ensuring robust object representations for accurate downstream tasks.

Head: Dual Label Assignments for NMS-Free Inference
The head of YOLOv10 introduces a dual label assignment strategy to replace traditional NMS, combining one-to-many assignments for richer supervision during training with one-to-one assignments for efficient inference by selecting the best prediction per object.

A consistent matching metric aligns these assignments, ensuring precise object detection and enabling an efficient, NMS-free deployment pipeline.

Matching Metric: The matching metric $m(\alpha,\beta)$ evaluates the agreement between predictions and ground-truth objects, combining classification and localization accuracy:

$$m(\alpha,\beta)=s\cdot p\alpha\cdot \mathrm{IoU}(b\hat{},b)\beta$$

s: Spatial prior indicating whether the prediction lies within the ground-truth object.
p: Classification confidence score.
IoU(b^,b): Intersection-over-Union between predicted bounding box b^ and ground-truth bounding box b.
α,β: Hyperparameters controlling the weight of classification and localization tasks.

Supervision Gap Analysis: The gap between the two branches is measured by the classification objectives they generate:

$$A=\mathrm{to2o},i - l(i\in\Omega)\mathrm{to2m},i + \sum k\in\Omega \mathrm{to2m},k$$

to2o,i: Classification target for the one-to-one branch.
to2m,i: Classification target for the one-to-many branch.
Ω: Set of positive samples for one-to-many matching.
l(i∈Ω): Indicator function for whether a prediction is in the positive set.

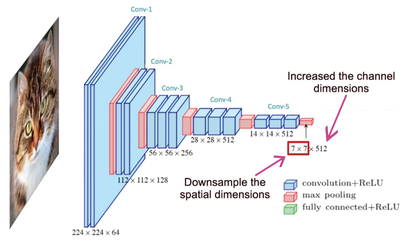
The consistent matching metric ensures to2o,i aligns closely with to2m,i, reducing A, the supervision gap.

Consistent Matching for Dual Branches: To ensure both branches (one-to-one and one-to-many) align well, YOLOv10 uses a consistent matching metric:
*mo*2o = *mo*2m*r*
mo2o: One-to-one matching metric.
*mo*2m: One-to-many matching metric.
r: Consistency factor, defaulted to *r*=1 for simplicity.
Using consistent metrics minimizes the supervision gap between the two branches, and the one-to-one branch (used during inference) benefits from the rich supervision of the one-to-many branch.

3.2. Efficiency-Accuracy Model Design

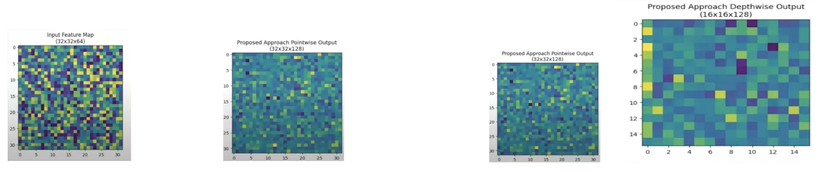
Lightweight Classification Head
The classification head employs **depthwise separable convolutions**, significantly reducing computational cost while maintaining high classification performance, minimizing the overhead associated with feature processing in the final detection stages, optimizing resource usage for real-time applications.

Spatial-Channel Decoupled Downsampling
When processing an image, models typically downsample spatial resolution and increase channel depth in a single step, which can lead to information loss. YOLOv10 separates these operations using a **pointwise convolution** for channel adjustment followed by a **depthwise convolution** for spatial downsampling. This decoupled approach retains critical information while reducing computational load, ensuring efficient feature extraction.



Current Method in YOLOs (3x3 Convolution with a stride of 2)
Pointwise convolution (1×1) adjusts the number of channels in the feature map while preserving the original spatial dimensions (H×W). For example, when an input feature map is processed through a pointwise convolution, the width and height remain unchanged, but the number of channels increases, such as expanding to 128 channels.

Depthwise convolution (3×3 with stride 2) performs spatial downsampling by reducing the spatial dimensions from H×W to H/2×W/2 while keeping the channel count unchanged. After a pointwise convolution, applying depthwise convolution reduces the spatial size from 32×32 to 16×16, with the channel count remaining constant.



3.3. Rank-Guided Block Design

The rank-guided block dynamically adjusts the complexity of each stage based on the intrinsic rank analysis of feature redundancy. Low-rank layers with redundant information use simpler operations to save computation, while high-rank layers with critical information retain complex blocks for accuracy. This approach optimizes resource usage and improves efficiency without sacrificing performance.

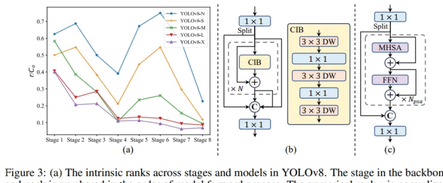
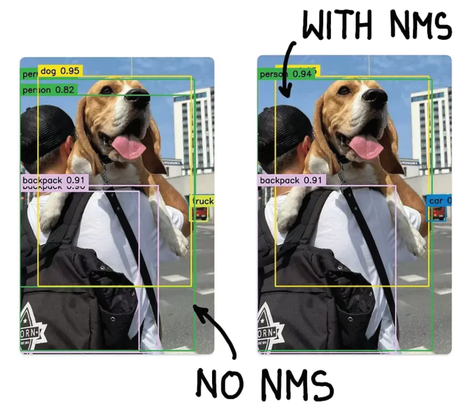


Figure 3: (a) The intrinsic ranks across stages and models in YOLOv8. The stage in the backbone and neck is numbered in the order of model forward process. The numerical rank r is normalized to r/C_o for y-axis and its threshold is set to $\lambda_{max}/2$, by default, where C_o denotes the number of output channels and λ_{max} is the largest singular value. It can be observed that deep stages and large models exhibit lower intrinsic rank values. (b) The compact inverted block (CIB). (c) The partial self-attention module (PSA).

4. RESULTS



YOLOv10 achieves state-of-the-art performance in real-time object detection, demonstrating significant improvements in speed and efficiency across benchmarks. On the COCO dataset, YOLOv10-S delivers a 1.8× speed improvement over RT-DETR-R18 while maintaining comparable Average Precision (AP), and YOLOv10-X matches RT-DETR-R101 in accuracy with a 1.3× faster inference speed. The model also reduces latency by 46% and parameters by 25% compared to YOLOv9-C, making it more suitable for resource-constrained systems. With six scalable variants (N, S, M, B, L, X), YOLOv10 adapts to a wide range of applications, from lightweight tasks to high-accuracy requirements. Furthermore, it achieves superior efficiency-accuracy trade-offs, outperforming models like YOLOv8, YOLOv9, and RT-DETR. These results establish YOLOv10 as a leading solution for real-time object detection.

5. CONCLUSION

YOLOv10 redefines real-time object detection by introducing an efficient, NMS-free pipeline and a scalable architecture that adapts to diverse application needs. Its innovations in efficiency-accuracy design and dual label assignments position it as a benchmark for future models. Moving forward, efforts to close performance gaps in smaller models and expand to multi-modal detection tasks will further enhance its versatility.

6. REFERENCES

Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., Ding, G. (2024). YOLOv10: Real-Time End-to-End Object Detection. Tsinghua University. [Scan the QR code for more details].

Redmon, J., Farhadi, A. (2016). YOLOv1: Unified, Real-Time Object Detection. arXiv.

Carion, N., et al. (2020). End-to-End Object Detection with Transformers (DETR). arXiv.