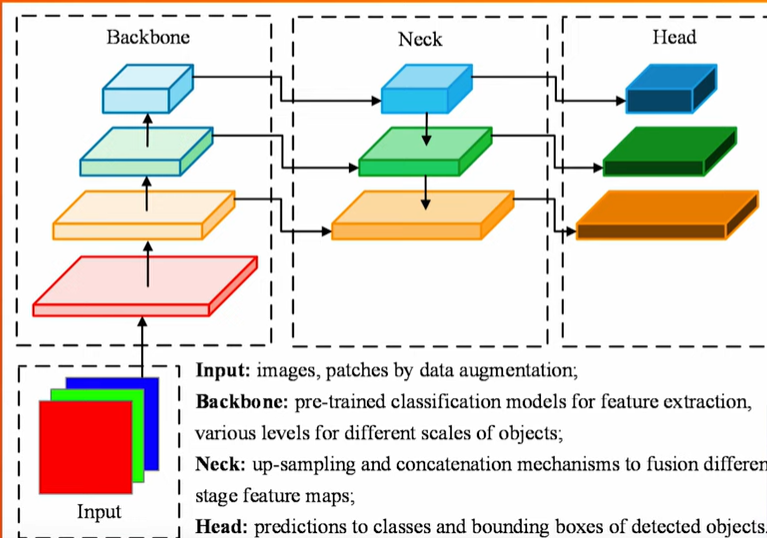
YOLO: có 3 phần chính:



Backbone: trích xuất đặc trưng (feature) từ input

* Nhận ảnh đầu vào, đi qua nhiều lớp convolution
* Mỗi lớp học ra các đặc trưng khác nhau: tầng thấp để học đường nét, màu sắc. Tầng cao học hình dạng, vật thể.

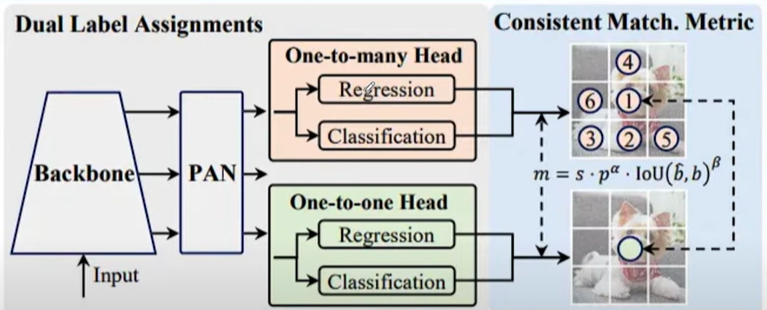
Neck: kết hợp đặc trưng từ các tầng của Backbone để tăng khả năng phát hiện ở nhiều điểm.

* FPN(Feature pyramid network): truyền thông tin từ thấp lên cao
* PAN (Path aggregation network): truyền thông tin từ cao xuống thấp

Head: dự đoán đầu ra

* Bounding box
* Class
* Confidence score

1. Dual label assignments



Yolov10 sử dụng 2 nhánh head khác nhau để học đặc trưng:

* One-to-many: một ground truth (GT) có thể được gán cho nhiều anchor, giúp học tổng quát hơn.
* One-to-one: mỗi GT chỉ được gánn cho một anchor duy nhất, giúp dự đoán chính xác và gọn gàng.

Việc chọn anchor dựa trên công thức Matching Score: 

Cả hai head được huấn luyện cùng lúc, cho phép phần backbone và neck của mô hình tận dụng được sự giám sát toàn diện từ việc gán nhãn one-to-many, từ đó cải thiện khả năng học và độ chính xác của mô hình.

Và sau đó, trong giai đoạn suy luận (inference), head one-to-many sẽ bị loại bỏ (giống như kỹ thuật "deep supervision"). Chỉ head one-to-one được sử dụng để đưa ra dự đoán. Cách tiếp cận này đảm bảo rằng mô hình có thể được triển khai end-to-end mà không gây thêm chi phí tính toán nào trong quá trình suy luận.

* Bỏ one-to-many -> chỉ dùng one-to-one -> không cần bước Non-max-suppression (NMS).

1. Rank-guided block design (thiết kế khối dẫn hướng theo thứ hạng)

Yolov10 đề xuất một phương pháp thiết kế khối theo thứ hạng nhằm giảm độ phức tạp của các tầng bị đánh giá là dư thừa, bằng cách sử dụng thiết kế kiến trúc gọn nhẹ.

Ý tưởng:

* Mỗi tầng trong backbone sẽ được đánh giá mức độ quan trọng (rank), sắp xếp theo thứ tự tăng dần.
* Những tầng ít quan trọng hơn (rank thấp, dư thừa), được thay bằng khối hiệu quả hơn có tên là CIB (Compact Invert Block).

Cách làm:

* Thay dần từng block từ tầng thấp lên.
* Nếu thay mà không giảm hiệu năng → tiếp tục.
* Nếu giảm hiệu năng → dừng lại.

Mô hình trở nên nhẹ hơn mà vẫn đảm bảo độ chính xác.

Ngoài ra, yolov10 sử dụng PAN làm neck vì PAN có khả năng truyền thông tin hiệu quả từ tầng cao xuống tầng thấp và ngược lại, giúp mô hình kết hợp được các đặc trưng từ nhiều cấp độ.

1. Lightweight classification & regression head
   1. Classifiction head (phân loại đối tượng):

* Object Identification (nhận dạng đối tượng): phần classification head chịu trách nhiệm xác định lớp của từng đối tượng được phát hiện trong mỗi khung bao (bounding box).

VD: nó có thể phân loại các đối tượng thành: người, xe hơi, chó,..

* Class probability estimation (ước lượng xác suất lớp): nó tính toán xác suất mà một đối tượng được phát hiện thuộc về mỗi lớp có thể xảy ra. Việc này bao gồm áp dụng hàm softmax để tạo ra xác suất cho nhiều lớp, đảm bảo tổng các xác suất bằng 1.
  1. Regression head:
* Bounding box prediction: phần regression head chịu trách nhiệm dự đoán tọa độ chính xác của các bounding box bao quanh đối tượng. Bao gồm việc dự đoán tọa độ trung tâm (x, y), chiều rộng, chiều cao của mỗi khung bao
* Box confidence score (điểm tin cậy của khung): cho biết độ chắc chắn khung chứa vật thể.
* Nghiên cứu cho thấy: regression head quan trọng hơn -> YOLOv10 làm classification head nhẹ hơn để tăng tốc mà không giảm hiệu năng.

Vì vậy họ áp dụng một kiến trúc nhẹ hơn cho classification head, bao gồm:

* Hai lớp convolution tách biệt theo chiều sâu (depthwise separable convolutions) (3x3).
* Một lớp convolution 1x1.

1. Accuracy driven design

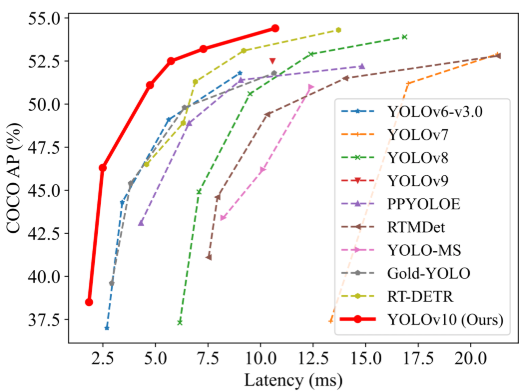
Large-Kernel convolution (tích chập với kernel lớn):

* Dùng kernel lớn -> mô hình có thể nhìn xa hơn, nắm bắt vật thể nhỏ hoặc mờ tốt hơn.
* Cải thiện khả năng phát hiện vật thể nhỏ trong ảnh.

Partial Self-Attention (PSA – cơ chế tự chú ý một phần):

* Attention nhẹ, giúp mô hình hiểu mối quan hệ xa trong ảnh mà không tốn quá nhiều tài nguyên.

1. Biểu đồ:

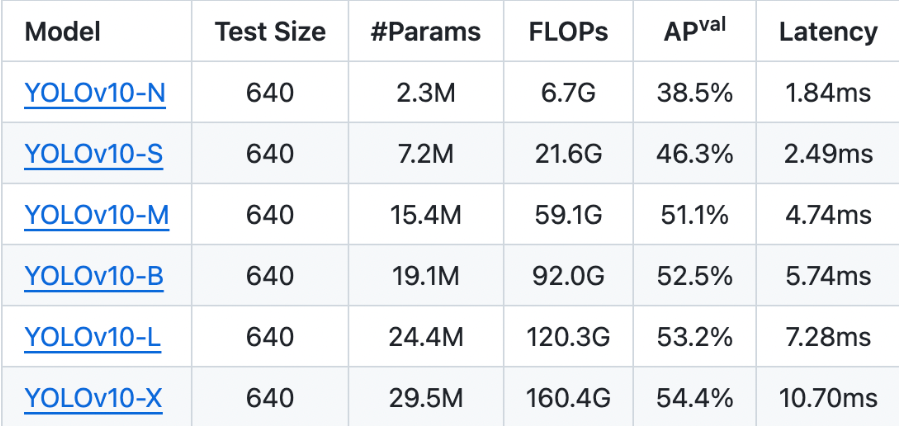


COOP AP (%): đo độ chính xác: càng cao thì mô hình càng chính xác

Latency (ms): thời gian xử lý: càng thấp thì mô hình càng nhanh

Tại cùng mức latency, Yolov10 luôn có AP cao hơn các mô hình khác. Tại cùng mức AP, yolov10 có latency thấp hơn.

1. Bảng thống kê của Yolov10:



Bao gồm: phiên bản, kích thước ảnh đầu vào, số lượng tham số học, số lượng phép tính (biểu thị độ phức tạp), độ chính xác trung bình, độ trễ suy luận (tốc độ xử lý).

Các phiên bản N->X là từ nhẹ đến mạnh, tốc độ giảm dần, độ chính xác tăng dần, phù hợp với từng nhu cầu:

* N, S: phù hợp thiết bị nhúng, điện thoại, camera real-time
* M, B: phù hợp ứng dụng cần cân bằng tốc độ & chính xác.
* L, X: dùng cho cloud/server, đòi hỏi chính xác cực cao.

N – Nano

S – Small

M – Medium

B – Base

L – Large

X – Extra-larga

1. Các yolov10 xử lý:
   1. Trường hợp quá sáng, bị chói

Large-Kernel conv sẽ mở rộng vùng nhìn, không phụ thuộc hoàn toàn vào vùng bị lóa, mà tận dụng thông tin từ vùng lân cận.

PSA sẽ giúp mô hình tự động bỏ qua vùng chói và tập trung vào những vùng rõ hơn.

* 1. Trường hợp thiếu sáng

PAN kết hợp đặc trưng từ tầng thấp, giữ lại các chi tiết biên đơn giản có thể còn nhìn thấy trong ảnh thiếu sáng.

PSA: nó không cố nhìn hết ảnh, mà tìm các khu vực nổi bật (vùng sáng nhẹ) để chú ý vào.

* 1. Trường hợp mờ

Large-Kernel conv học được dạng khối mờ, không cần chi tiết rìa (Yolov10 đã được huấn luyện trên ảnh bị blur mô phỏng).

PSA giúp tìm lại vị trí trung tâm của vật thể.

* 1. Trường hợp sương mù

PSA nhận diện hình thể chung của vật thể, không dựa vào chi tiết.

Một số anchor vẫn học được vật thể, dù bị làm mờ nhẹ.