TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN, ĐHQG-HCM

KHOA KHOA HỌC MÁY TÍNH

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
VNUHCM - UIT

BÁO CÁO ĐỒ ÁN MÔN HỌC

ĐỀ TÀI: NHẬN DIỆN XE Ô TÔ BẰNG CÁC THUẬT TOÁN HỌC SÂU (YOLOV8n VÀ YOLOV11n)

Môn học: CS331.P11 - THỊ GIÁC MÁY TÍNH NÂNG CAO

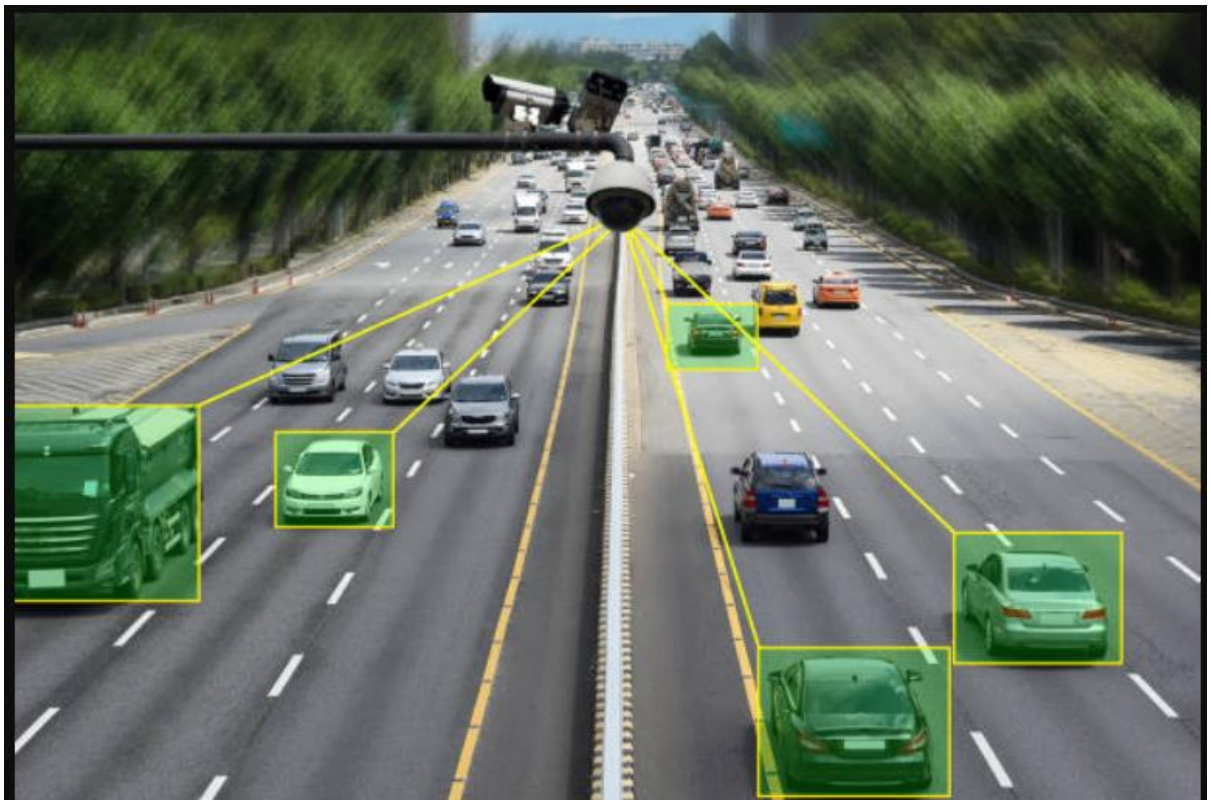
Giảng viên hướng dẫn: TS Mai Tiến Dũng

Thực hiện bởi: Trần Giang Sử - 22521266

TÓM TẮT

Việc xác định các phương tiện giao thông càng ngày càng yêu cầu độ chính xác cao hơn và cho hiệu suất tốt hơn đối với môi trường thực tiễn (bài toán thời gian thực). Qua đồ án này, em muốn so sánh hiệu năng giữa các phiên bản YOLO, cụ thể là YOLOv8 và YOLOv11, đặc biệt là khi YOLOv11 vừa mới được ra mắt vào tháng 9/2024. Bài toán hướng đến việc nhận diện xe ô tô (phân loại đúng và xác định đúng

vị trí - bounding box của đối tượng), xem xét xem liệu các cải tiến ở phiên bản YOLOv11 có tăng được hiệu suất bài toán, đặc biệt là việc xác định chính xác được các đối tượng nhỏ.



Chương I. TỔNG QUAN BÀI TOÁN.

1. Giới thiệu bài toán.

1.1. Lý do và tầm quan trọng của bài toán.

"Trong những năm gần đây, sự phát triển của công nghệ thị giác máy tính (computer vision) đã tạo ra những đột phá đáng kể trong nhiều lĩnh vực, từ y tế, nông nghiệp đến giao thông vận tải. Một trong những ứng dụng quan trọng của thị giác máy tính là **nhận diện đối tượng** (object detection), đặc biệt là **nhận diện xe ô tô**.

Lý do:

- **An toàn giao thông:**
    - Hệ thống hỗ trợ lái xe: Phát hiện và cảnh báo va chạm, hỗ trợ đỗ xe tự động, điều khiển hành trình thích ứng.
    - Giám sát giao thông: Đếm xe, phân loại loại xe, phát hiện các hành vi vi phạm luật giao thông.
- **Quản lý đô thị:**
    - Điều khiển giao thông thông minh: Tối ưu hóa tín hiệu đèn giao thông, quản lý bãi đỗ xe.
    - Phân tích lưu lượng giao thông: Đánh giá tình hình giao thông, lập kế hoạch phát triển hạ tầng.
- **An ninh:**
    - Giám sát an ninh: Phát hiện các đối tượng khả nghi, theo dõi các phương tiện.
- **Nghiên cứu:**
    - Phát triển các thuật toán mới trong lĩnh vực thị giác máy tính.
    - Ứng dụng trong các lĩnh vực khác như robot, thực tế ảo.

Tầm quan trọng:

- **Đáp ứng các nhu cầu thực tế:** Nhận diện xe ô tô là bài toán có ý nghĩa thực tế cao, góp phần giải quyết các vấn đề liên quan tới đời sống, đặc biệt là về vấn đề giao thông
- **Góp phần vào sự phát triển của khoa học công nghệ:** Việc nghiên cứu và phát triển các thuật toán nhận diện xe ô tô hiệu quả sẽ thúc đẩy sự phát triển của ngành công nghiệp ô tô, giao thông vận tải và các ngành công nghiệp liên quan.
- **Mở ra cơ hội phát triển nhiều ứng dụng khác trong thực tế:** Kết quả nghiên cứu có thể được ứng dụng trong nhiều lĩnh vực khác nhau, tạo ra các sản phẩm và dịch vụ mới.

1.2 Mục tiêu bài toán hướng đến

- **Đánh giá hiệu suất của các mô hình YOLOv8 và YOLOv11:** So sánh độ chính xác, tốc độ xử lý, khả năng xác định chuẩn xác các đối tượng nhỏ và khó phát hiện hơn
- **Tối ưu hóa mô hình:** Điều chỉnh các siêu tham số, tìm kiếm cấu trúc mạng phù hợp để nâng cao hiệu suất của mô hình.
- **Xây dựng một hệ thống nhận diện xe ô tô thực thời:** Hệ thống có thể xử lý video từ camera giao thông với tốc độ khung hình cao, đáp ứng yêu cầu của các ứng dụng thực tế.

2. Phát biểu bài toán.

2.1. Input của bài toán

Đầu vào của bài toán bao gồm 2 phần:

- Tập dữ liệu (dataset) bao gồm:

- File định dạng dữ liệu: data.yaml
  - Vì bài toán tập trung việc xác định một đối tượng là xe ô tô, vì vậy số number_class (nc) sẽ được thay đổi thành 1
- Tập train: 300 images + 300 labels
- Tập validation: 58 images + 58 labels
- Các images đã đều được gán nhãn bằng labelImg, phần mềm hỗ trợ gán nhãn cho dữ liệu
- Các file labels được lưu theo định dạng YOLO trong file .txt, tên file label tương ứng với tên file image, dữ liệu lưu theo dạng:

```
<class_id> <x_center> <y_center> <width> <height>
```

- Dữ liệu trên được thu thập từ camera giao thông (qua các phần mềm camera giao thông trực tuyến)

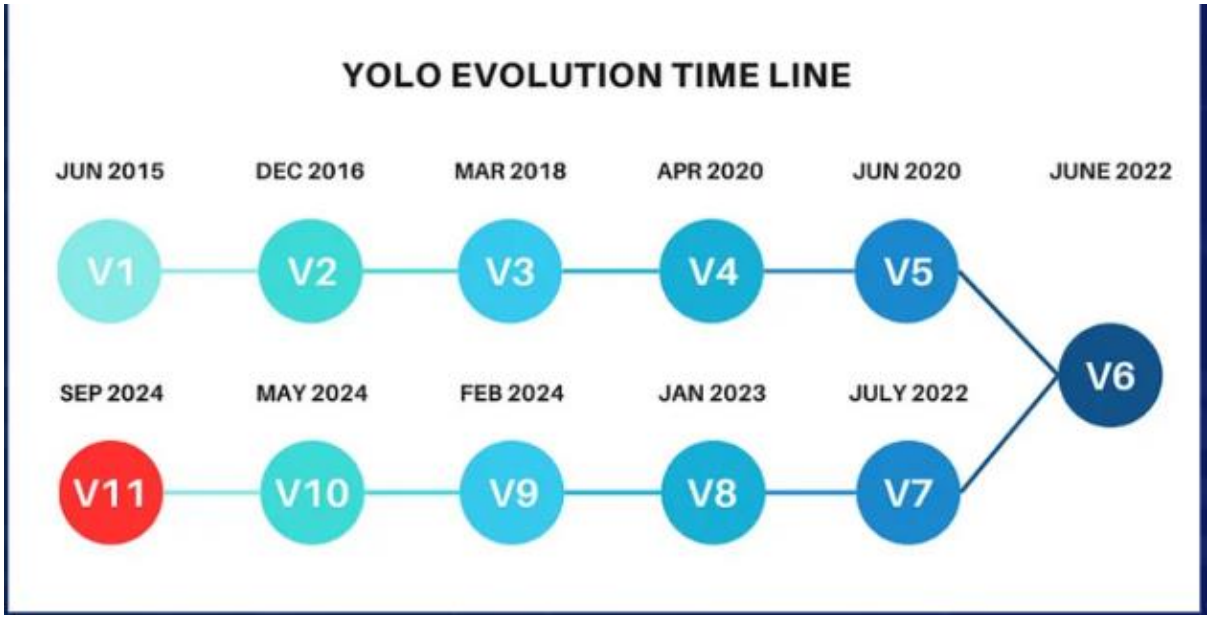
- Video/image cần xác định đối tượng

2.2. Output của bài toán

Đầu ra của bài toán này là video/image với các đối tượng (ô tô) đã được xác định và vẽ bounding box quanh các đối tượng. Ta có thêm lựa chọn nếu muốn đếm số lượng đối tượng có trong khung hình.

Chương II. PHƯƠNG PHÁP BÀI TOÁN

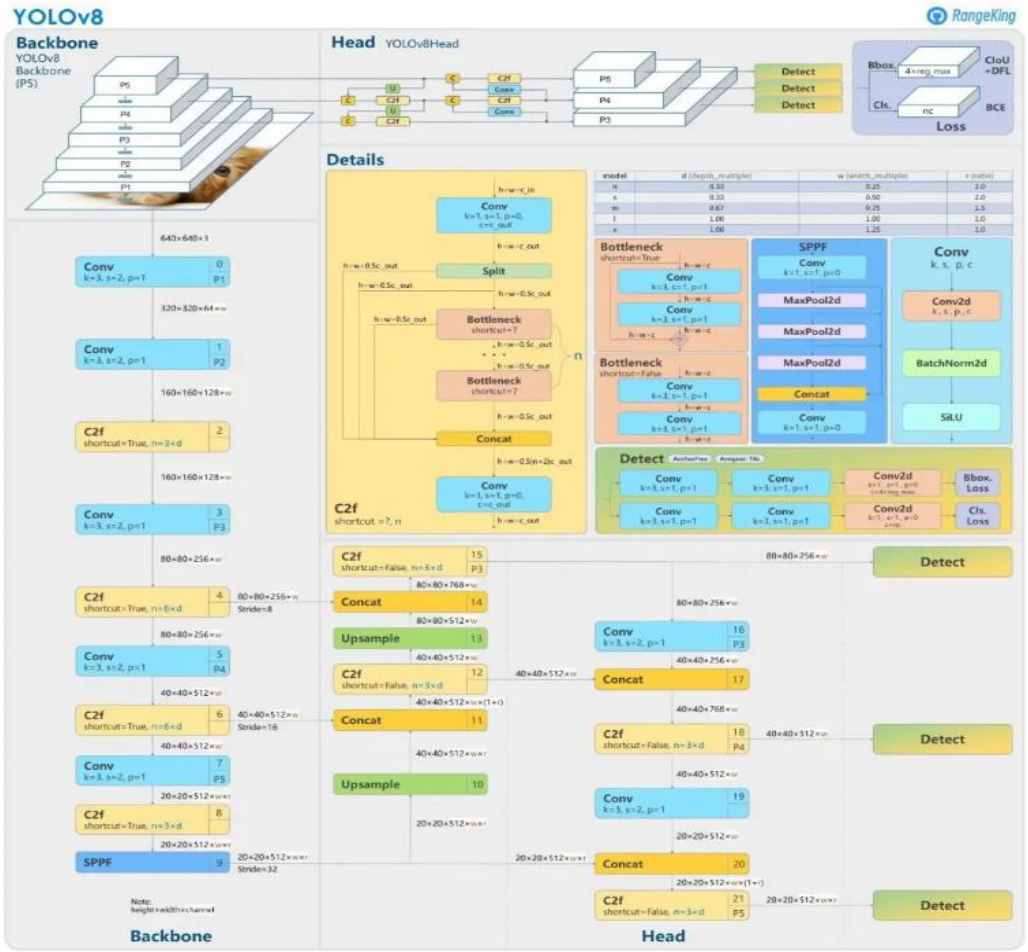
1. Giới thiệu về YOLO



- **YOLO (You Only Look Once)** là một thuật toán học sâu (deep learning) cực kỳ hiệu quả trong việc phát hiện đối tượng trong hình ảnh. Thay vì phân tích từng phần nhỏ của hình ảnh một cách độc lập như nhiều thuật toán khác, YOLO xem xét toàn bộ hình ảnh một lần duy nhất để xác định các đối tượng và vị trí của chúng. Điều này giúp YOLO có tốc độ xử lý cực kỳ nhanh, phù hợp với các ứng dụng đòi hỏi thời gian thực như video giám sát, tự lái xe.
- Với kiến trúc mạng thần kinh độc đáo, YOLO chia hình ảnh thành lưới các ô (grid). Mỗi ô sẽ dự đoán các bounding box (hộp bao quanh) và xác suất của các lớp đối tượng có thể xuất hiện trong ô đó. Nhờ vậy, YOLO có khả năng phát hiện nhiều đối tượng cùng lúc với độ chính xác cao.
- **Ưu điểm nổi bật của YOLO:**
    - **Tốc độ:** YOLO xử lý hình ảnh nhanh gấp nhiều lần so với các thuật toán khác, nhờ vào việc chỉ cần duyệt qua mạng thần kinh một lần.
    - **Đơn giản:** Cấu trúc của YOLO tương đối đơn giản, dễ dàng triển khai và tùy chỉnh.
    - **Khả năng phát hiện nhiều đối tượng:** YOLO có thể phát hiện nhiều đối tượng khác nhau trong một hình ảnh, kể cả các đối tượng nhỏ hoặc chồng lấp lên nhau.

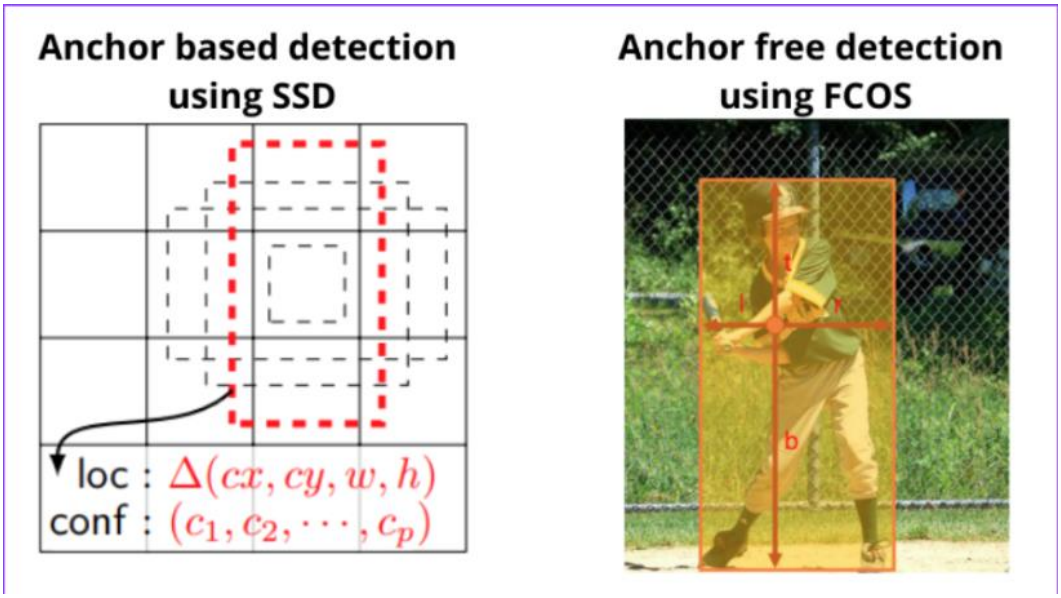
2. Giới thiệu về YOLOv8

- Theo timeline, YOLOv8 ra mắt vào đầu năm 2023, có những bước nhảy tiên tiến hơn so với các phiên bản trước (từ v7 đổ xuống).



Hình 1. Kiến trúc của YOLOV8.

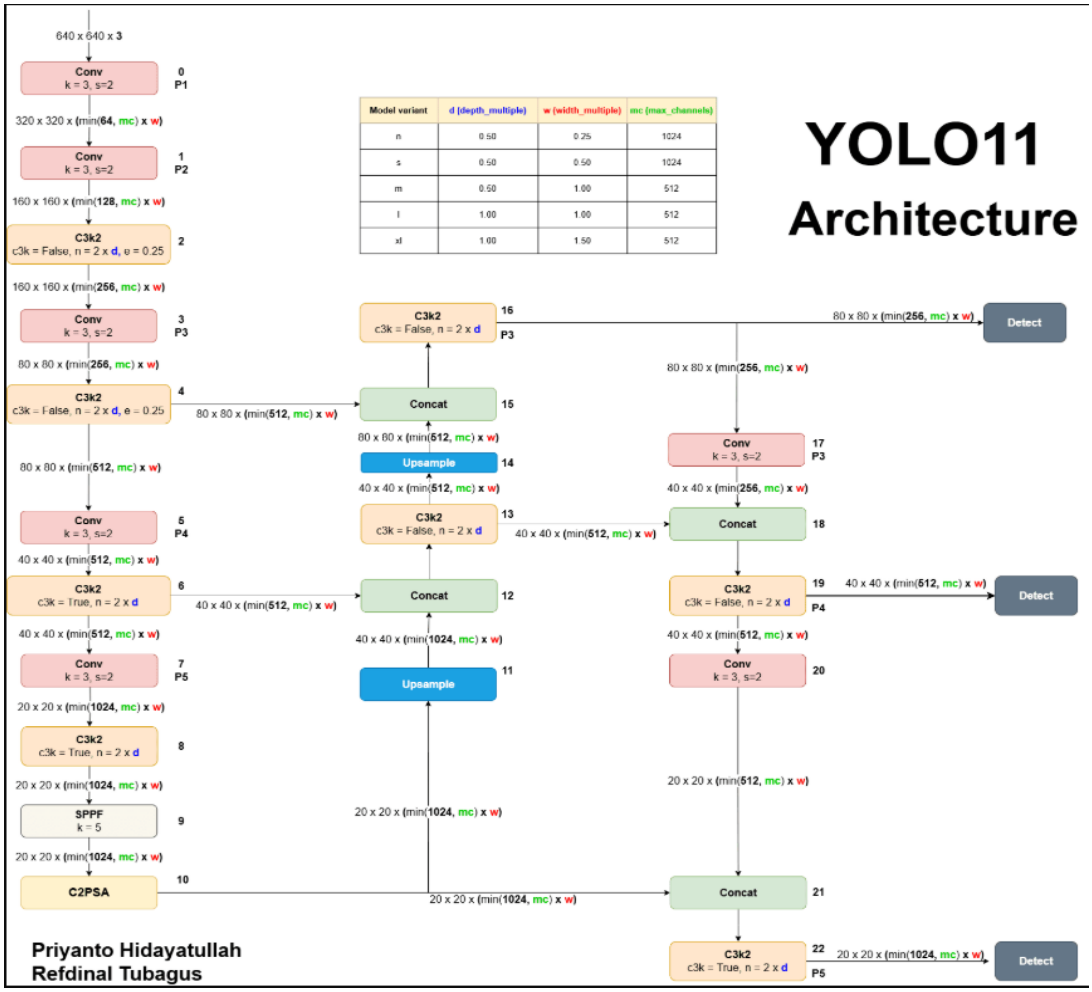
- Đến với phiên bản YOLOv8, cơ chế để dự đoán bounding box của các đối tượng đã thay đổi:
    - Ở các phiên bản trước, các mô hình vẫn sử dụng pre-defined box (hay còn gọi là anchor boxes) để xác định bounding box của các đối tượng (Đây gọi là anchor based detection)
    - Sang đến phiên bản YOLOv8, việc xác định bounding box của các đối tượng được thay đổi, không còn sử dụng các anchor boxes để tìm ra vị trí của đối tượng nữa, thay vào đó, một phương pháp mới được gọi là anchor free detection:



- Như tên gọi, phương pháp này không còn sử dụng các anchor boxes để tìm vị trí đối tượng nữa
- Mô hình lúc này sẽ sử dụng điểm tâm đối tượng, sau đó xem xét các mối quan hệ với các pixel lân cận để tìm ra bounding box của đối tượng, điều này làm tăng tốc độ học trên các bộ custom datasets.
- Đồng thời với phương pháp này, số lượng bounding boxes được dự đoán giảm đáng kể, giúp tăng tốc độ thuật toán Non-max Suppression (NMS) - thuật toán giúp loại bỏ các bounding boxes dự đoán sai.

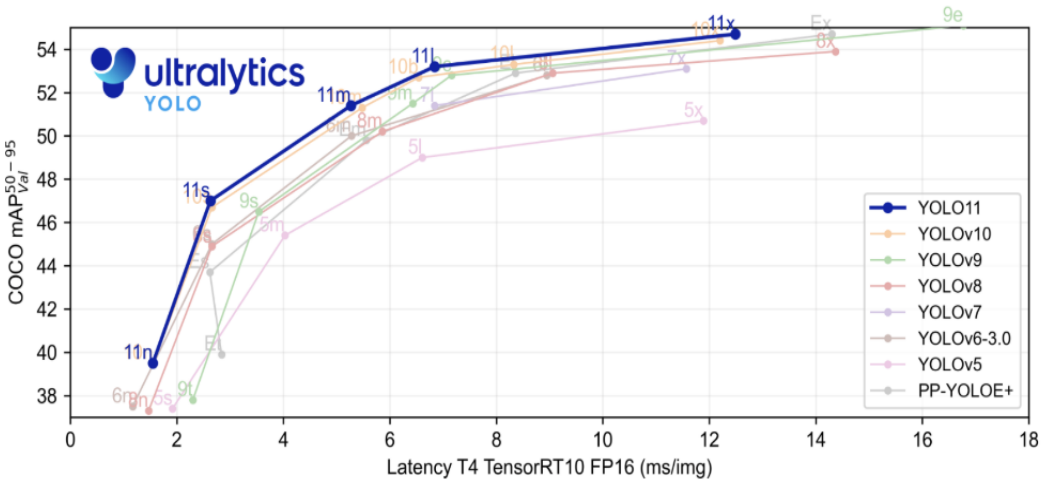
- Với phiên bản YOLOv8, vẫn có sự xuất hiện của layer C2F, nhưng với các tham số và số lượng các lớp lặp, kernel có thể đã được tối ưu so với các phiên bản trước. Vì vẫn có sự xuất hiện của layer C2F, nên tốc độ học tập sẽ tăng với các cấu hình đa nhân đa luồng, khi layer này có đặc điểm là chia đặc trưng ra và học song song ở 2 luồng, khiến cho mô hình sẽ hiểu đặc trưng tốt hơn.

3. Giới thiệu về YOLOv11

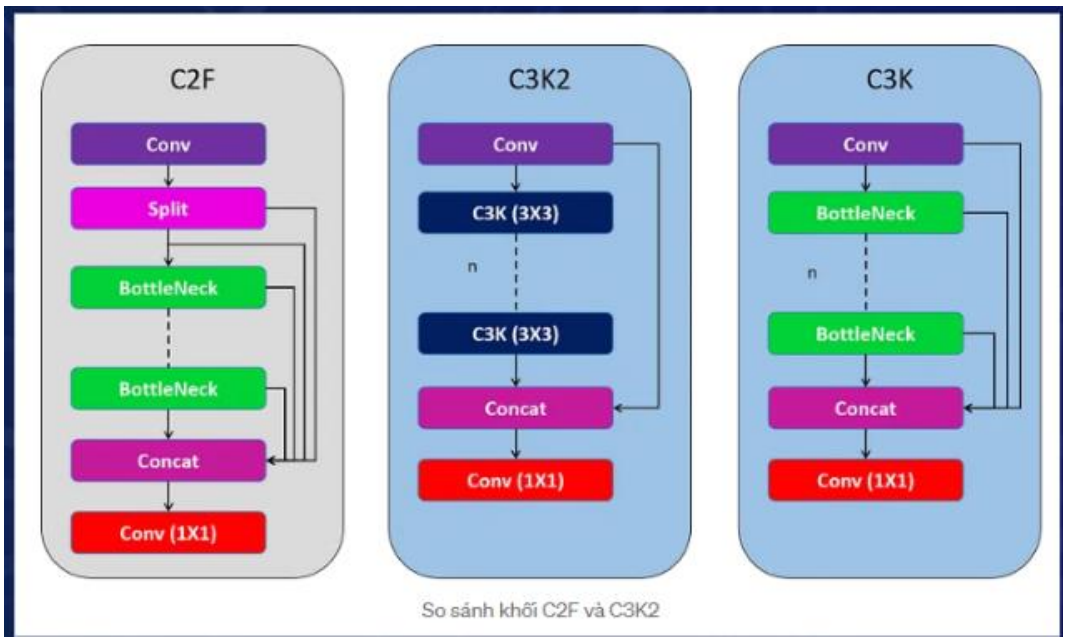


2. Kiến trúc của YOLOv11.

- YOLOv11 được giới thiệu vào tháng 9/2024. So với phiên bản YOLOv8, YOLOv11 đã có ít nhiều những sự thay đổi về mặt cấu trúc, dẫn đến việc khác biệt ở hiệu suất và độ chính xác:
    - Độ chính xác được cải thiện với độ phức tạp được giảm bớt: YOLOv11m đạt được điểm mAP vượt trội trên tập dữ liệu COCO trong khi sử dụng ít hơn 22% tham số so với biến thể YOLOv8m



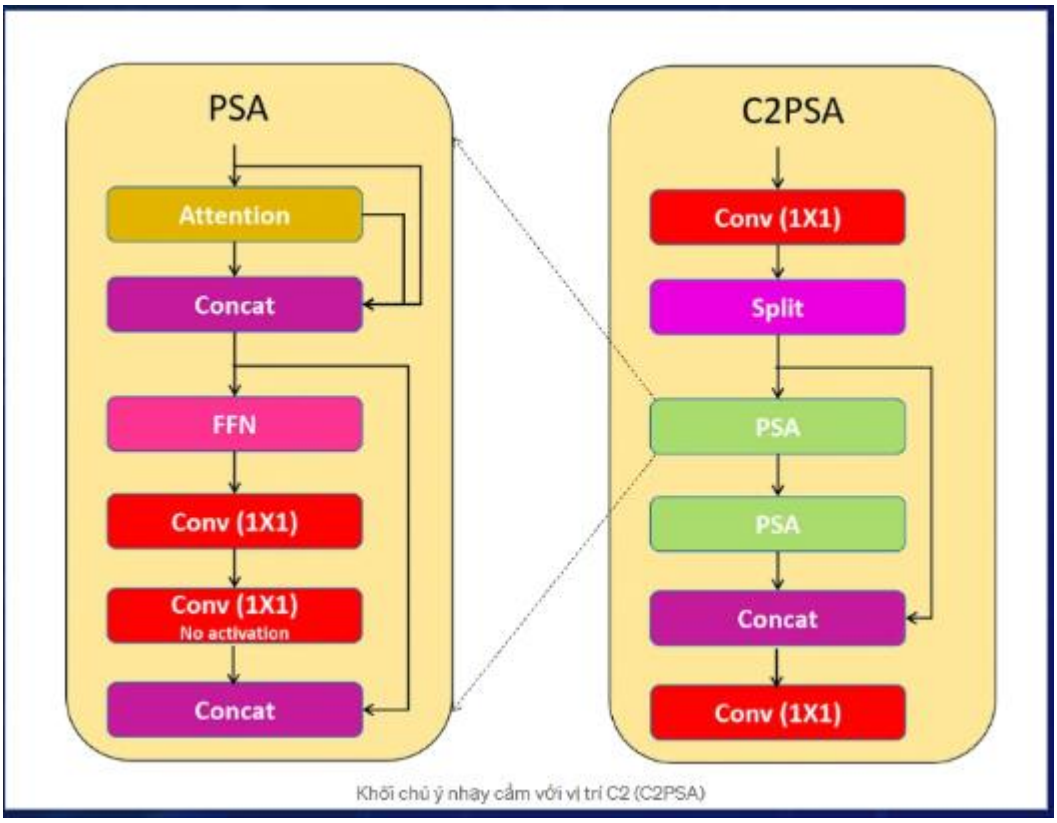
- So với YOLOv8, layer C3K2 đã thay thế vị trí của layer C2F, đây là layer đã có sự cải tiến ở các kiến trúc xương sống và cổ, qua đó tăng cường khả năng trích xuất đặc trưng, nhưng đồng thời cũng không còn việc học song song, nên nếu sử dụng các cấu hình đa nhân thì tốc độ sẽ không còn tăng, nhưng tổng quát thì YOLOv11 vẫn sẽ cho tốc độ học tập và xử lí nhanh hơn YOLOv8 trong nhiều điều kiện



So sánh khối C2F và C3K2

- Đặc biệt hơn, ở YOLOv11 có sự xuất hiện của C2PSA, Tính năng này cho phép mô hình tập trung hiệu quả hơn vào các vùng quan trọng trong hình ảnh, tăng cường khả năng phát hiện và phân tích các đối tượng. Khả năng chú ý được cải thiện đặc biệt có lợi cho việc xác định các đối tượng phức tạp hoặc bị che khuất một phần, giải quyết một thách thức phổ biến trong

các tác vụ phát hiện đối tượng.



4. Transfer learning

- (Transfer Learning) là một kỹ thuật trong học máy, nơi một mô hình đã được huấn luyện trên một tập dữ liệu lớn (thường là một nhiệm vụ) được sử dụng làm điểm khởi đầu để giải quyết một nhiệm vụ khác, thường với một lượng dữ liệu nhỏ hơn. Nói cách khác, thay vì bắt đầu huấn luyện một mô hình hoàn toàn mới từ đầu, chúng ta tận dụng kiến thức đã học được từ một mô hình đã được huấn luyện trước đó.
- Với bài toán này, lượng dữ liệu khá hạn hẹp, nên ta có thể sử dụng phương pháp transfer learning với các file weight (.pt) đã có sẵn của Ultralytics, qua đó tăng cường hiệu suất, tiết kiệm thời gian và tài nguyên huấn luyện hơn.
- Trong bài toán này chủ yếu tập trung tới việc đánh giá 2 model YOLOv8 và YOLOv11, vậy nên ta sẽ sử dụng các file weight nano (YOLOv8n và YOLOV11n), đặc điểm của chúng là nhẹ, nhanh (vì ít tham số).

Chương III. CÁC ĐỘ ĐO ĐÁNH GIÁ

1. Precision.

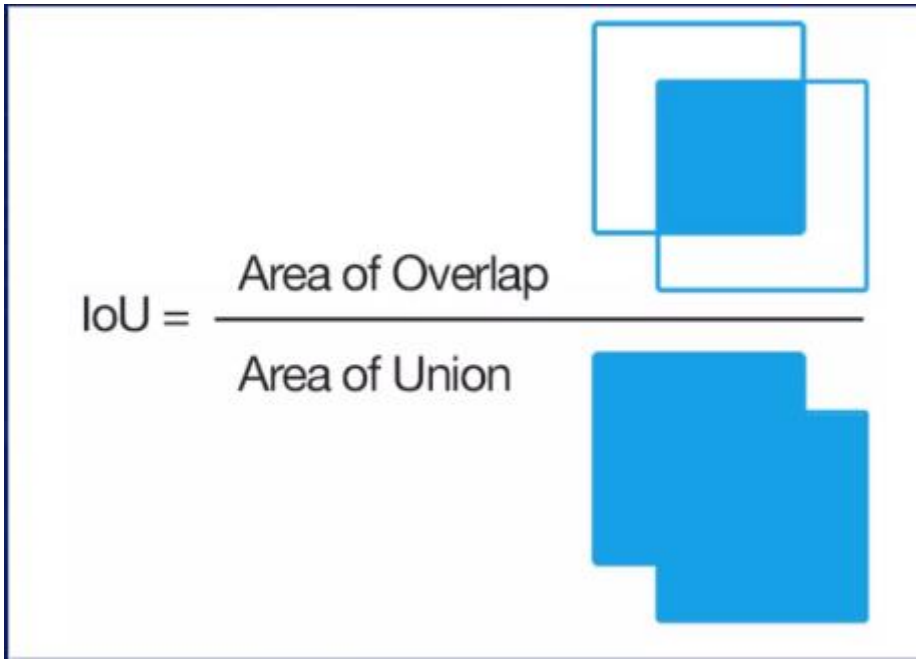
- **Precision = TP / (TP + FP)**
  - TP(True Positive): Bounding box dự đoán đúng lớp của đối tượng là ô tô
  - FP(False Positive): Không phải ô tô nhưng bounding box dự đoán lại là ô tô
- Precision phản ánh:
  - Precision cao: mô hình ít dự đoán nhầm các vật thể khác là xe ô tô
  - Precision thấp: nhiều vật thể không phải xe ô tô bị nhận nhầm là xe ô tô

2. Recall

- **Recall = TP / (TP + FN)**
  - TP(True Positive): Bounding box dự đoán đúng lớp của đối tượng là ô tô
  - FN(False Negative): Là ô tô nhưng bounding box dự đoán không phải là ô tô
- Recall phản ánh:
  - Recall cao: mô hình có khả năng nhận diện được hầu hết các xe ô tô thực tế có trong ảnh, ít bỏ sót xe.
  - Recall thấp: mô hình bỏ sót nhiều xe ô tô, không nhận diện được.

3. IoU (Intersection over Union)

- **Intersection over Union (IoU):** Đây là một thước đo để so sánh mức độ trùng khớp giữa một bounding box (hộp bao quanh đối tượng) do mô hình dự đoán và bounding box thực tế (ground truth). IoU được tính bằng tỉ lệ diện tích phần giao nhau giữa hai bounding box chia cho diện tích phần hợp của chúng.



- Area of Overlap: diện tích phần giao nhau giữa predicted bounding box với grouth-truth bouding box
- Area of Union: diện tích phần hợp giữa predicted bounding box với grouth-truth bounding box

4. AP@50

- AP là một chỉ số đo lường hiệu suất của mô hình ở các ngưỡng recall khác nhau.
- AP@50 có nghĩa là chúng ta chỉ xem xét các trường hợp mà IoU giữa bounding box dự đoán và ground truth lớn hơn hoặc bằng 50%. Điều này có nghĩa là, để một dự đoán được coi là đúng, phần chồng lấp giữa bounding box dự đoán và bounding box thực tế phải chiếm ít nhất 50% diện tích của một trong hai bounding box đó.

5. AP@50-95

- Thay vì chỉ xét tại một ngưỡng IoU duy nhất là 50%, AP50-95 tính trung bình các giá trị AP tại nhiều ngưỡng IoU khác nhau, từ 0.5 đến 0.95, với bước nhảy là 0.05.

- AP50-95 cho phép đánh giá một cách chính xác hơn khả năng của mô hình trong việc phát hiện các đối tượng ở các mức độ khó khác nhau.

Chương IV. KẾT QUẢ THỰC NGHIỆM.

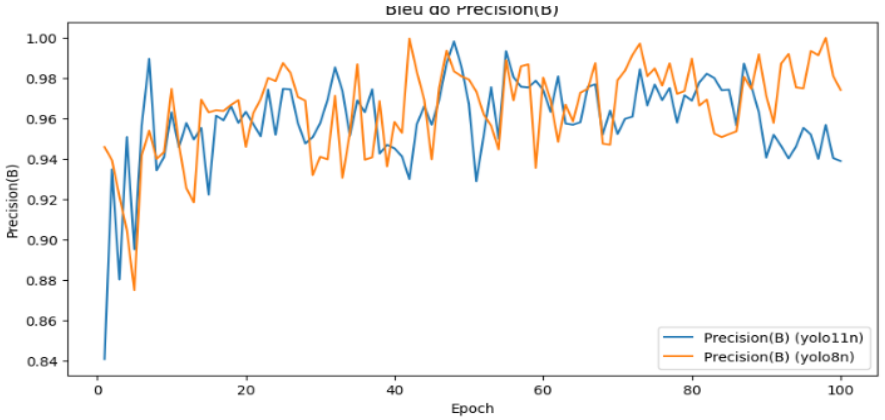
1. Xác định tham số

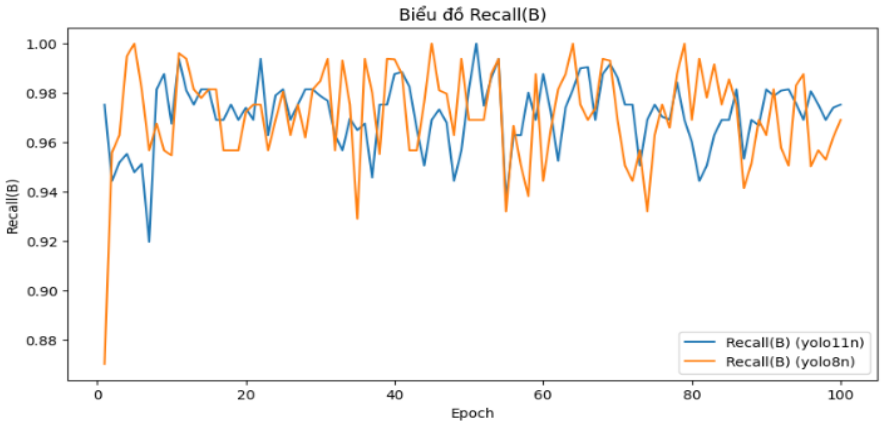
- Batch_size: 2, 4, 8
- Optimizer: SGD, AdamW
- Learning_rate: 1e-2, 1e-3, 1e-5
- Epoch: 100
- Training với file pre-trained weight của cả 2 phiên bản đều là bản nano (nhẹ và nhanh)

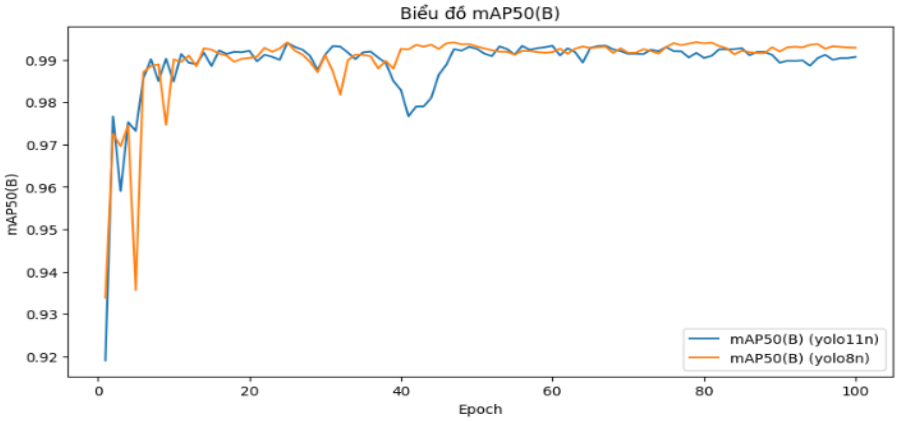
2. Kết quả huấn luyện

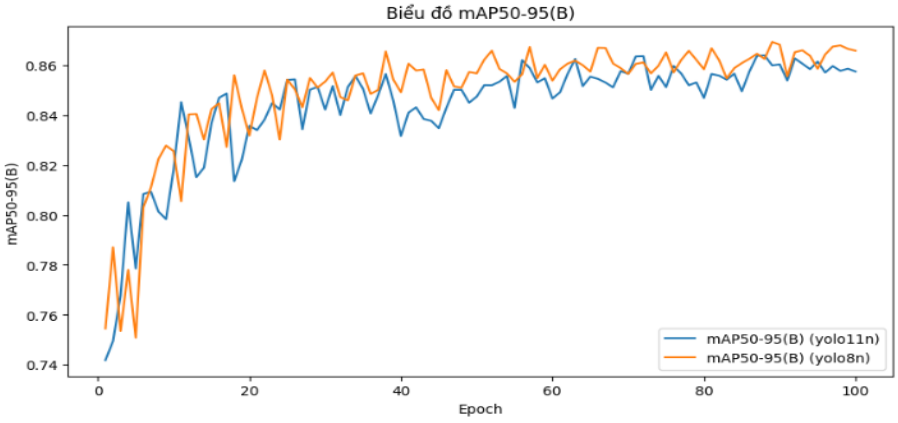
- Với mỗi hình, ta sẽ so sánh kết quả huấn luyện của 1 bộ tham số trên 2 mô hình
- Sau khi kiểm nghiệm và so sánh, ta rút ra được các bộ giá trị tham số tốt nhất:
    - Bs_SGD_0.001: Với batch_size tăng dần, thời gian huấn luyện sẽ giảm, đồng thời chỉ số map50-95 cũng sẽ giảm theo, có sự trade-off giữa độ chính xác và thời gian
    - Bs_AdamW_0.01: Cũng như cặp tham số trên, có sự trade-off giữa độ chính xác và thời gian huấn luyện
    - Nhưng với bộ dataset nhỏ nên thời gian huấn luyện không đáng kể
    - -> Giá trị tốt nhất cho bs là 2 với bài toán trên

1. Bs_optimizer_epoch_lr : 2_SGD_100_0.01

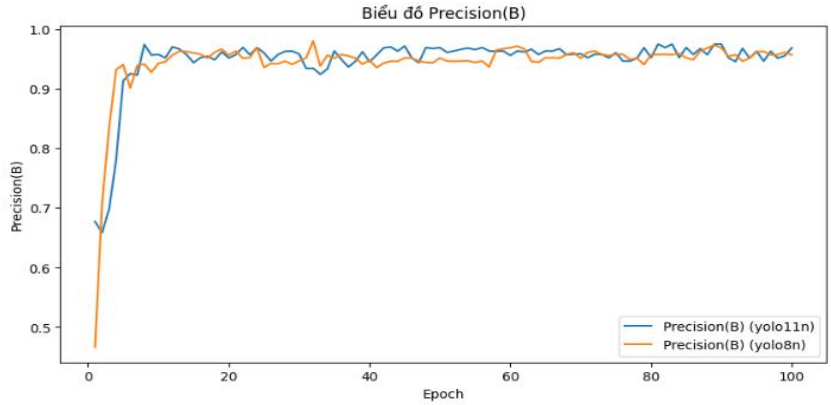


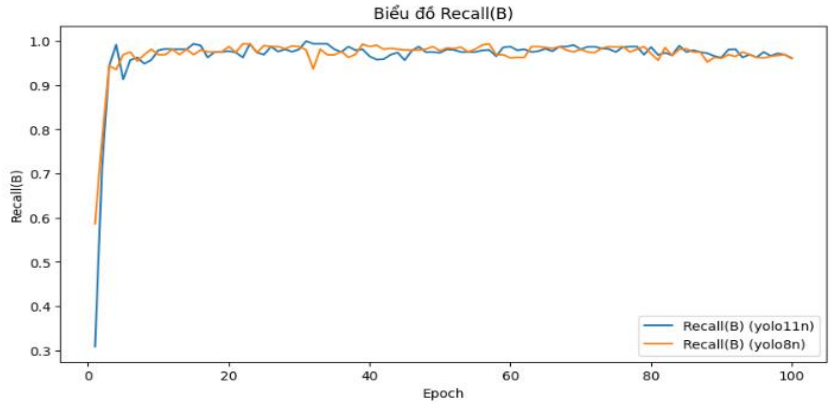


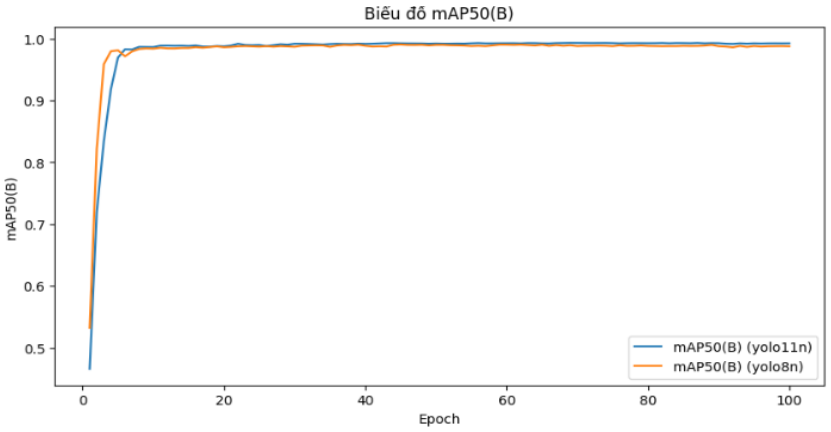


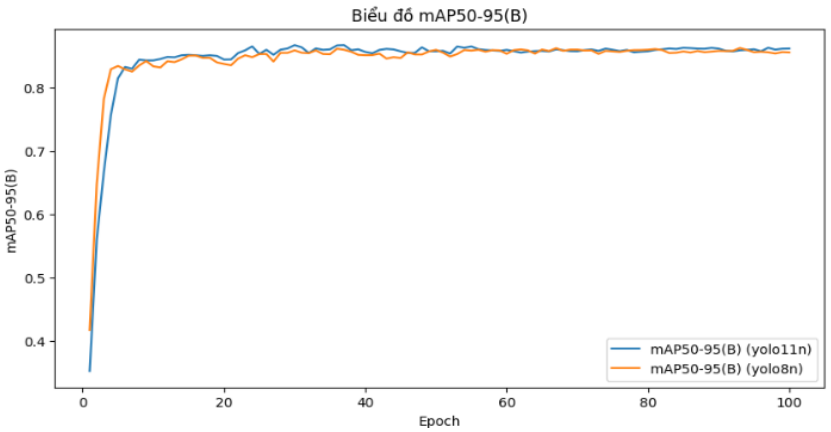


2. Bs_optimizer_epoch_lr : 2_SGD_100_0.001

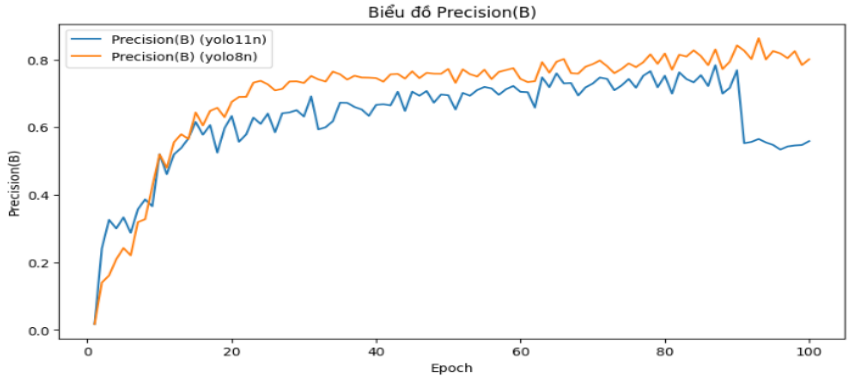


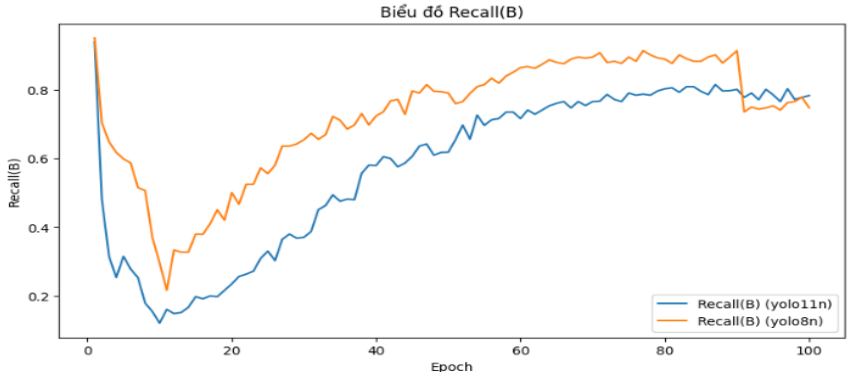


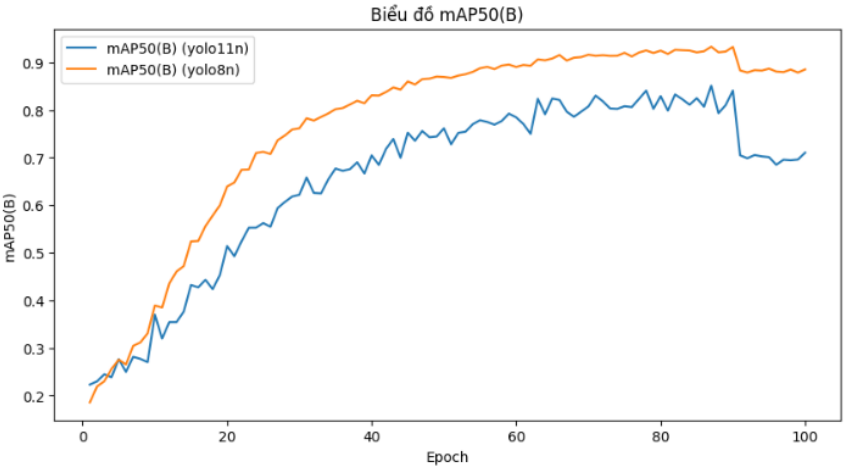


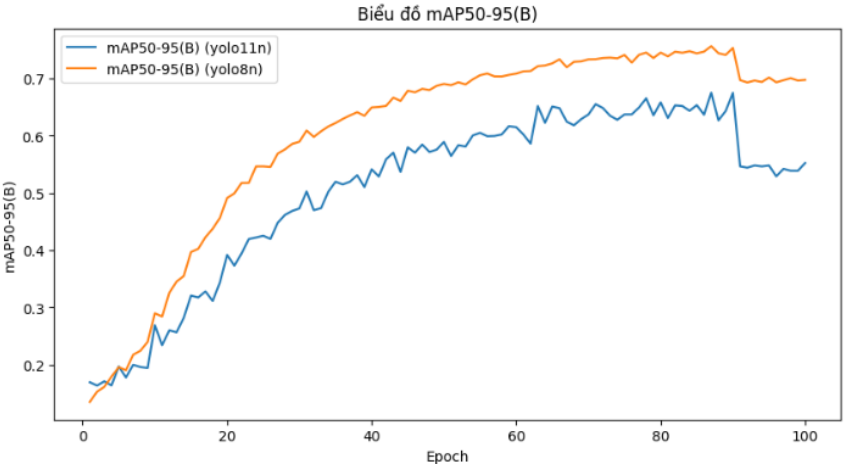


3. Bs_optimizer_epoch_lr: 2_SGD_100_1e-5

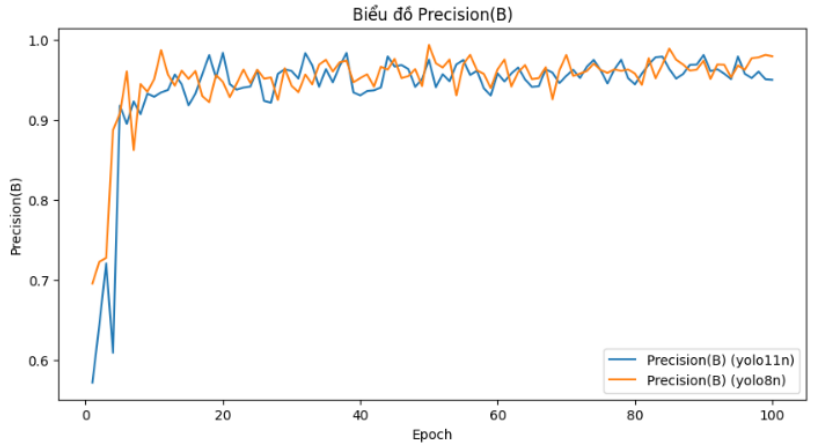


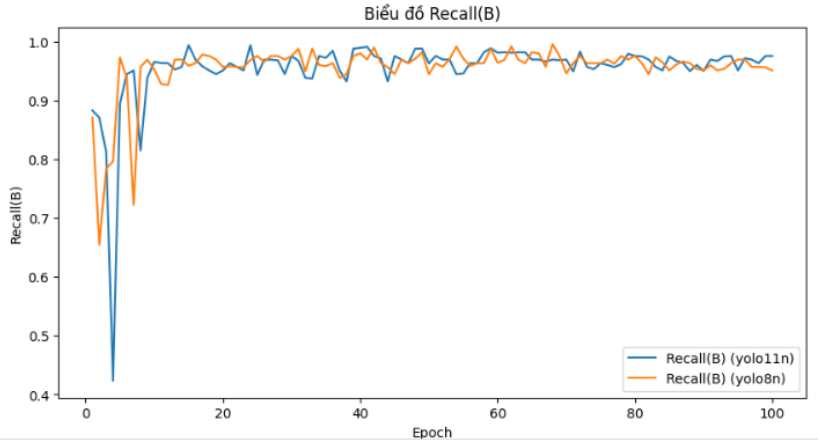


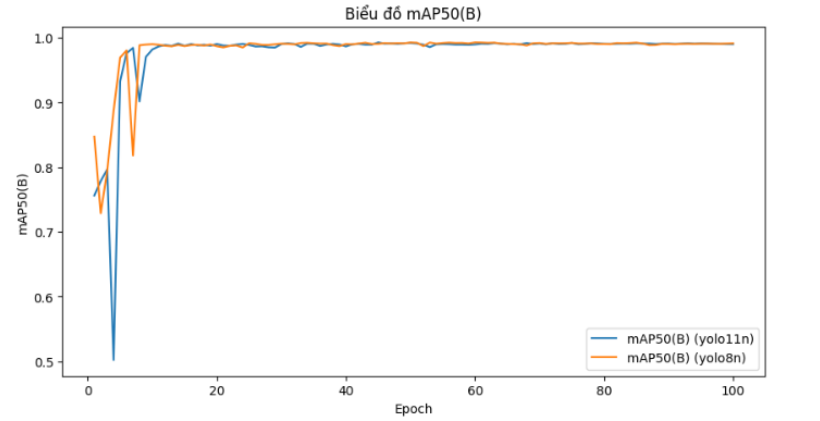


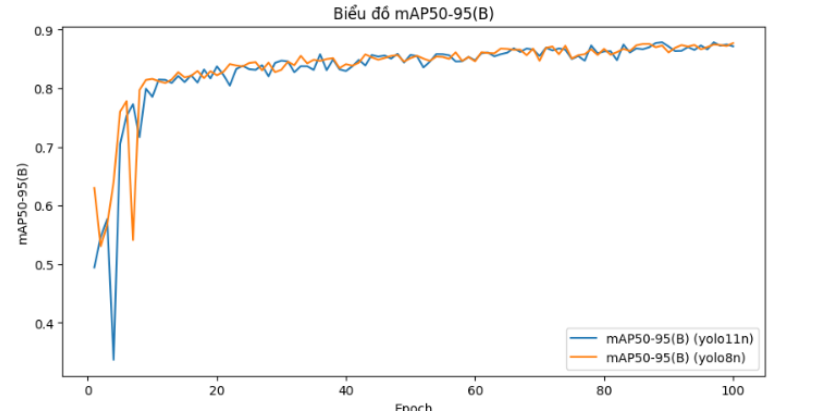


4. Bs_optimizer_epoch_lr: 2_AdamW_100_0.01

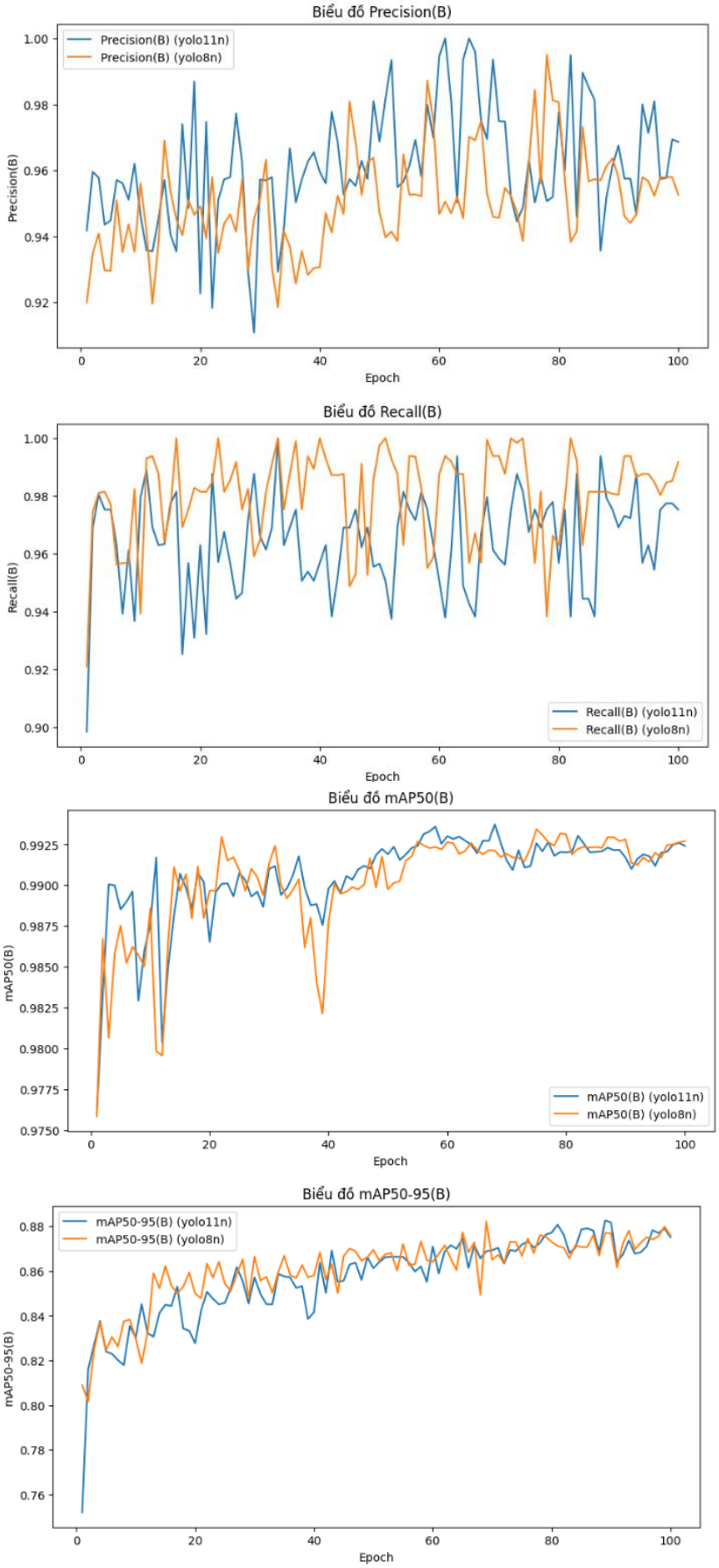




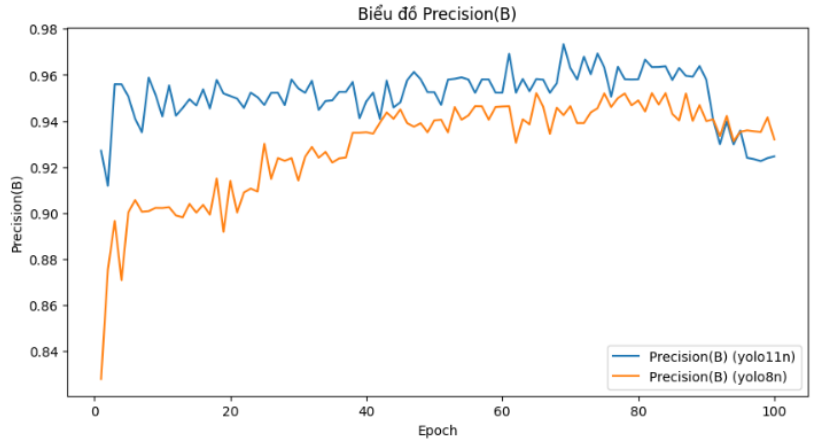


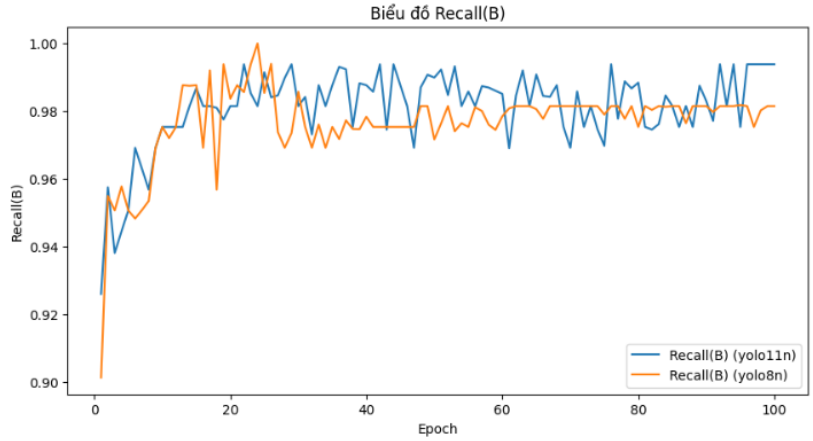


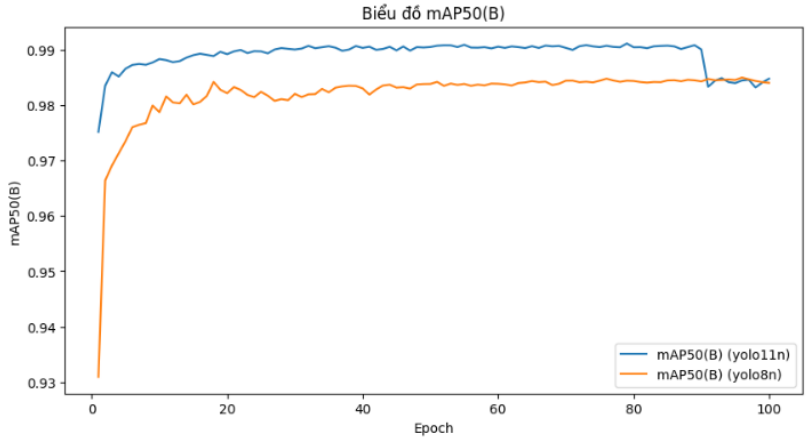
5. Bs_optimizer_epoch_lr: 2_AdamW_100_0.001

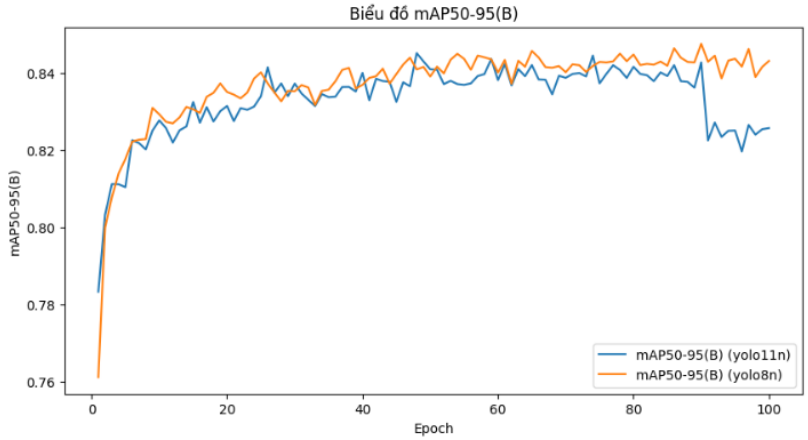


6. Bs_optimizer_epoch_lr: 2_AdamW_100_1e-5

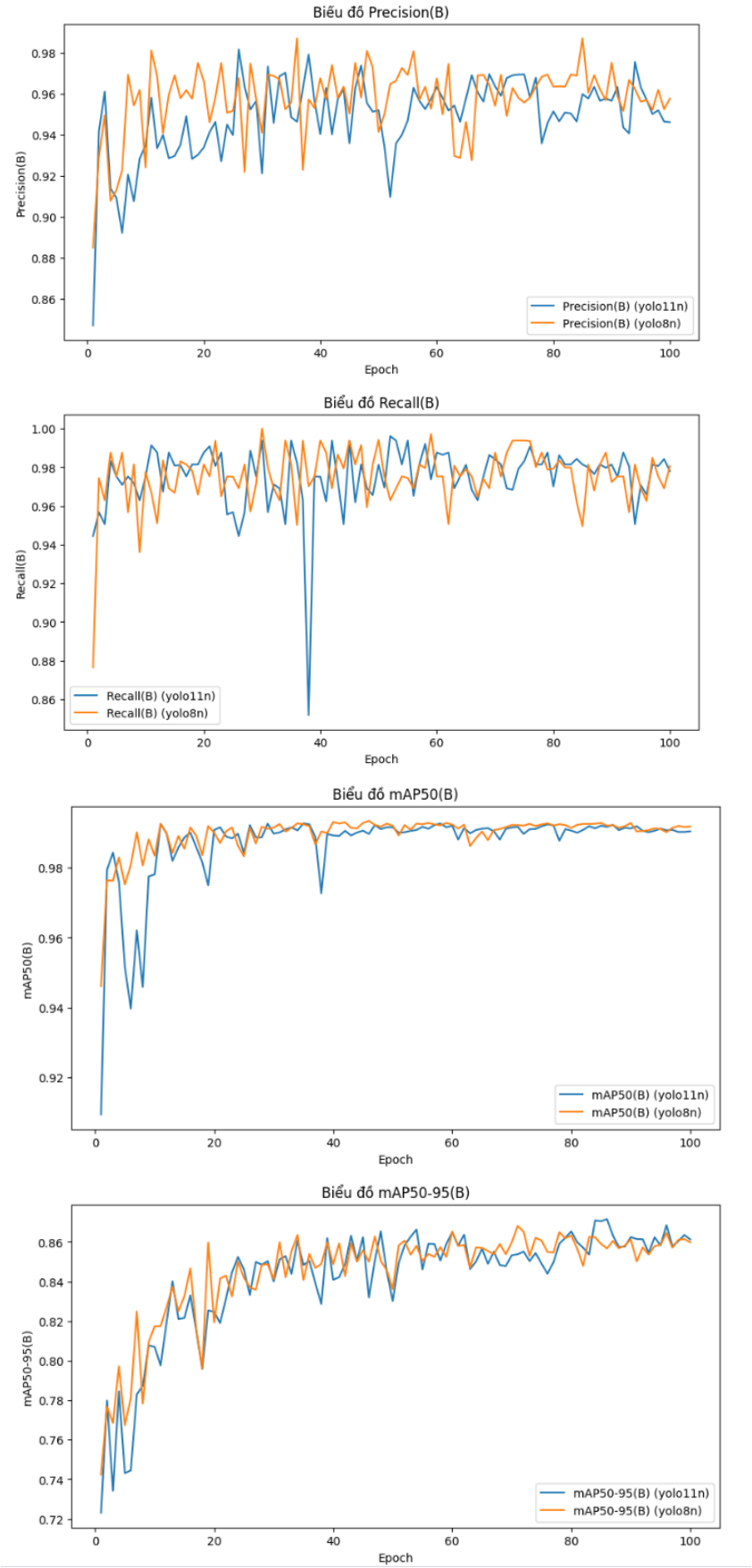




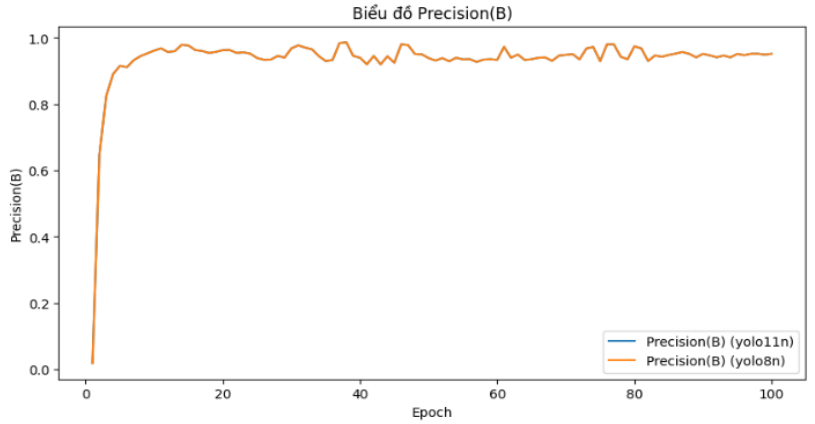


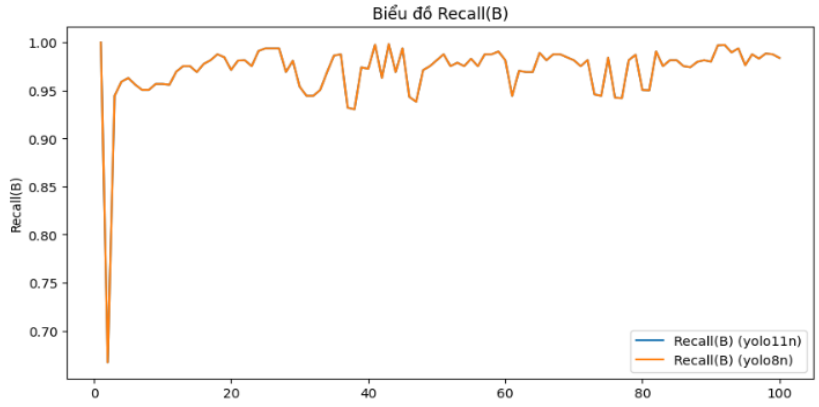


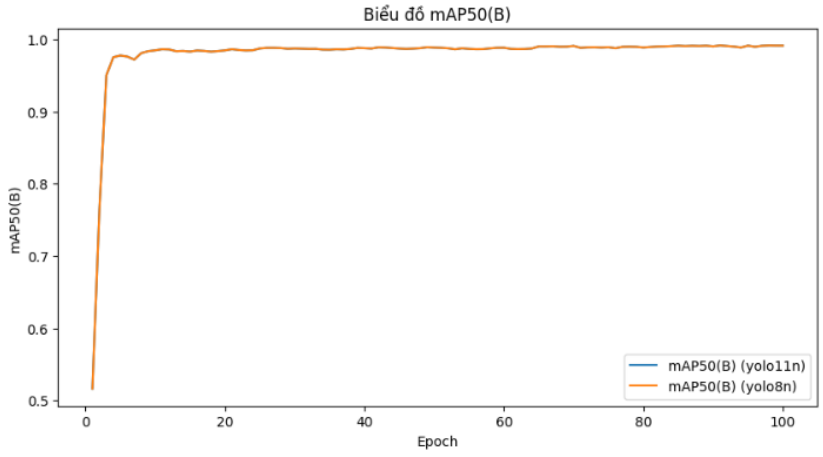
7. Bs_optimizer_epoch_lr: 4_SGD_100_0.01

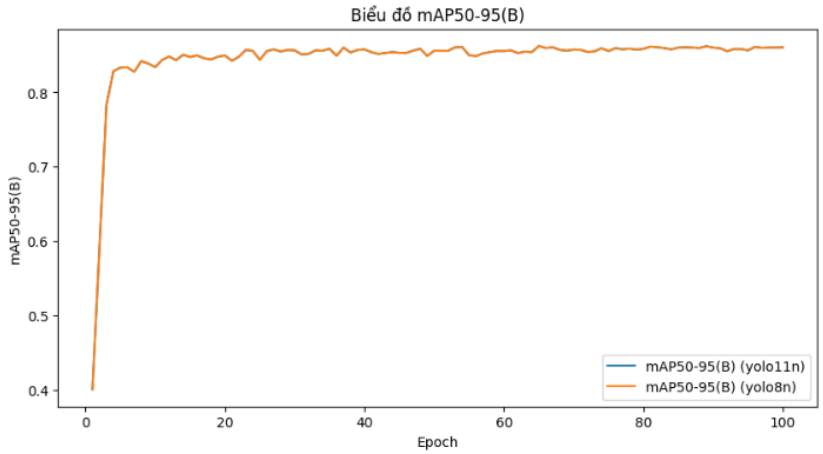


8. Bs_optimizer_epoch_lr: 4_SGD_100_0.001

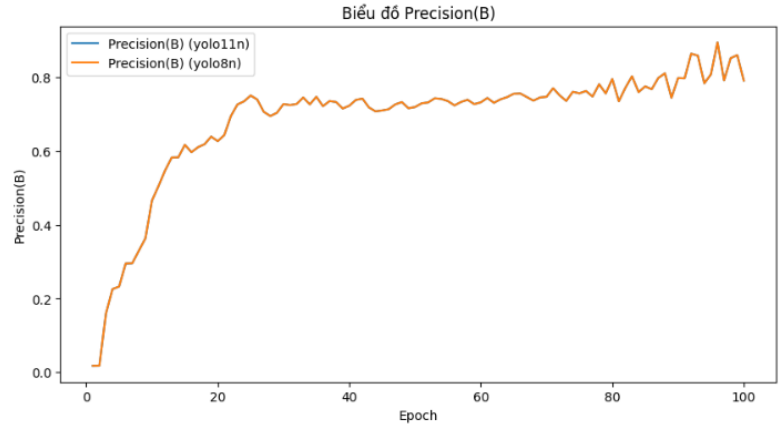


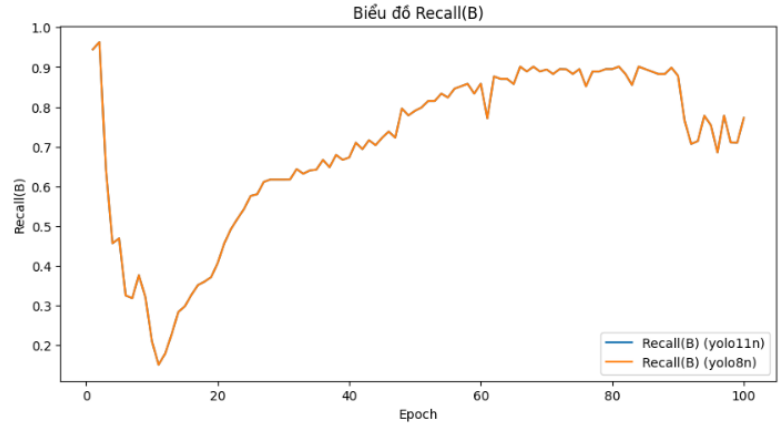


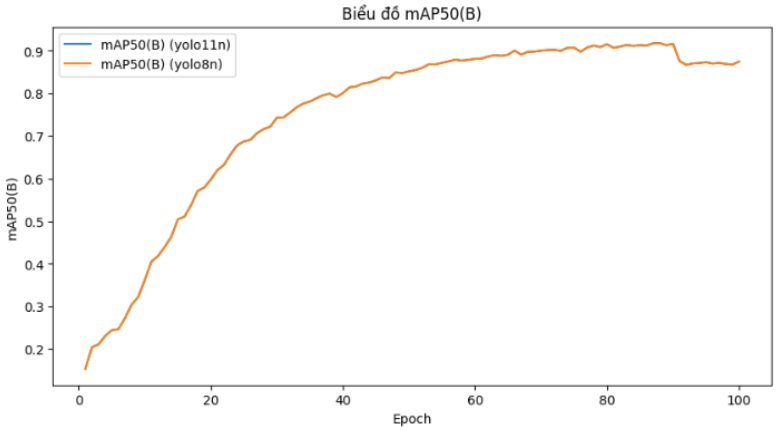


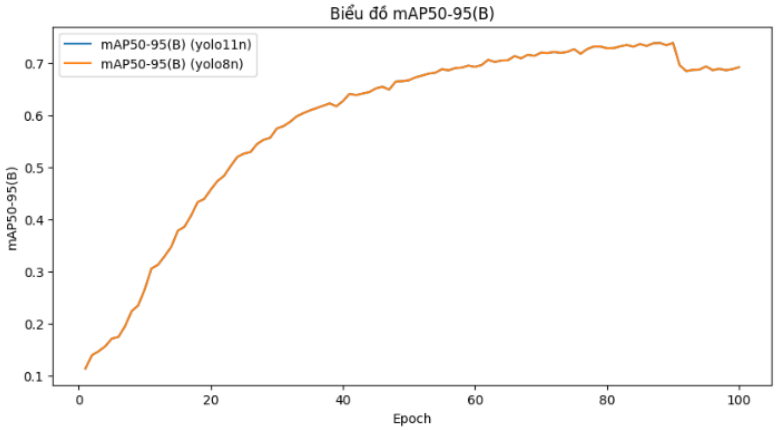


9. Bs_optimizer_epoch_lr: 4_SGD_100_1e-5

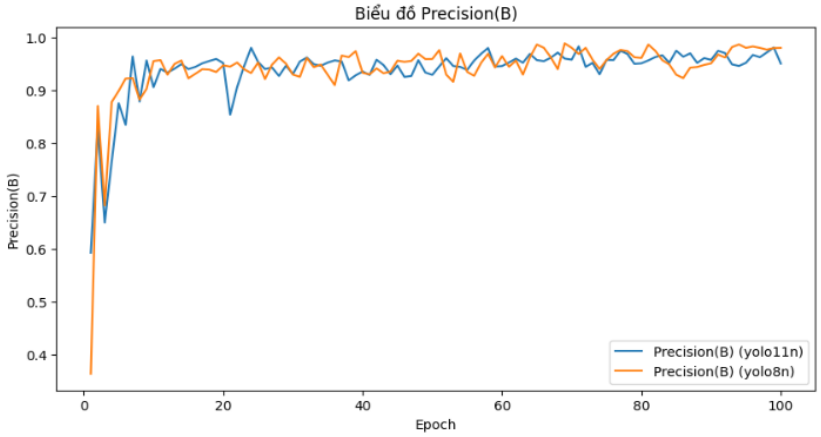


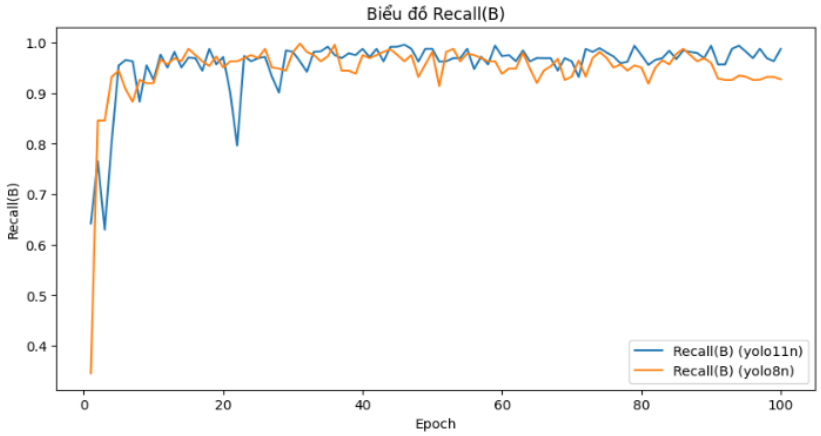


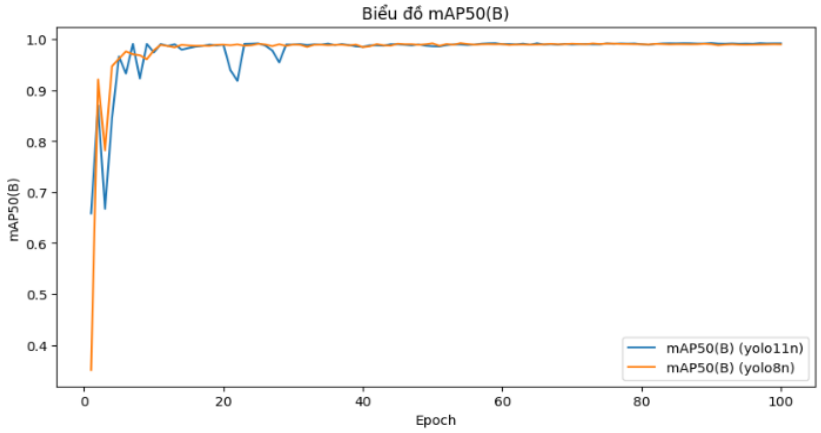


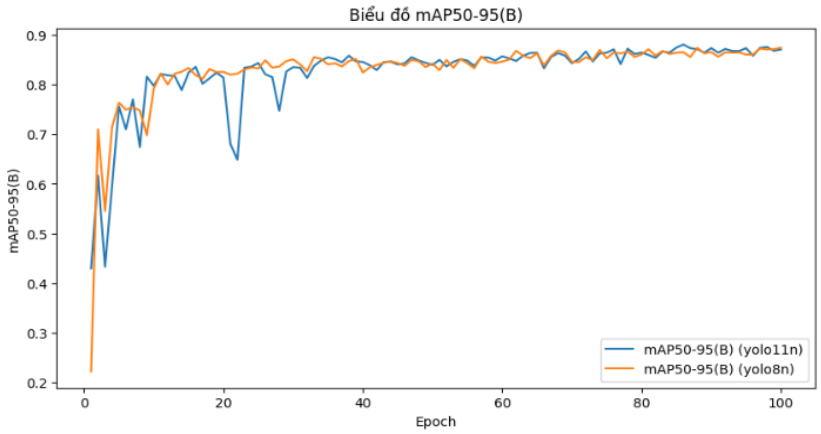


10. Bs_optimizer_epoch_lr: 4_AdamW_100_0.01

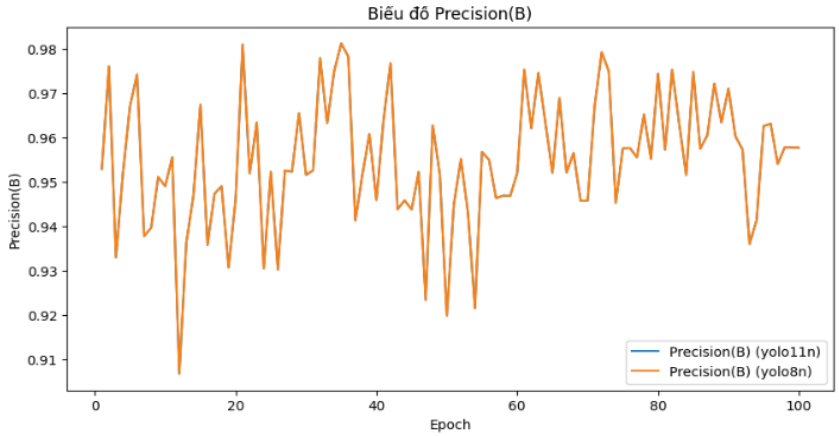


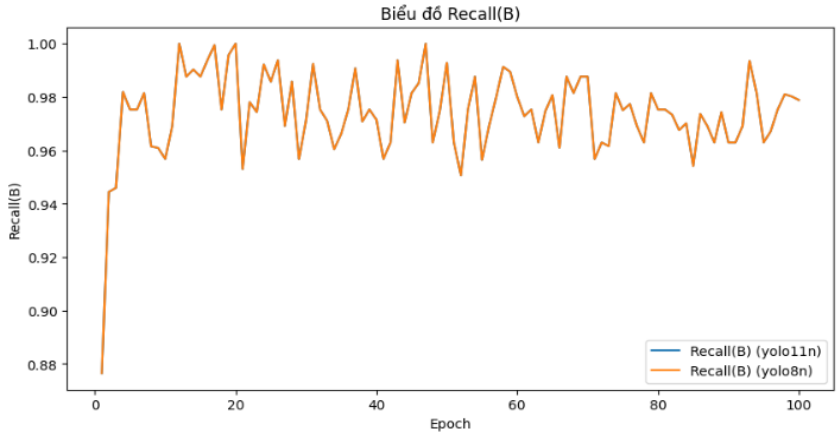


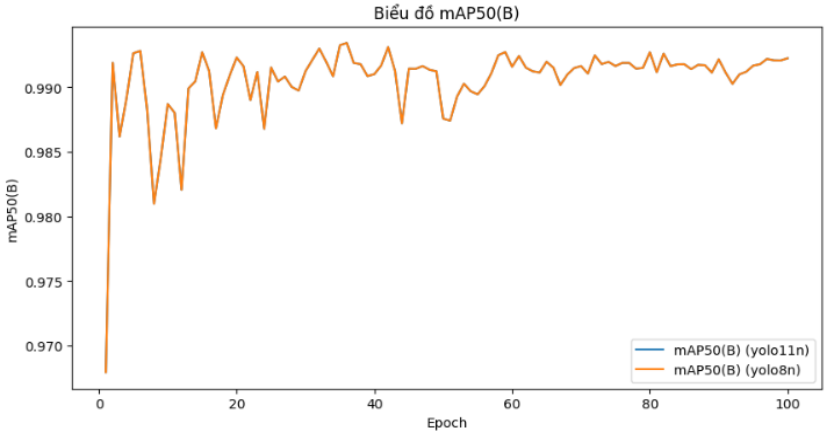


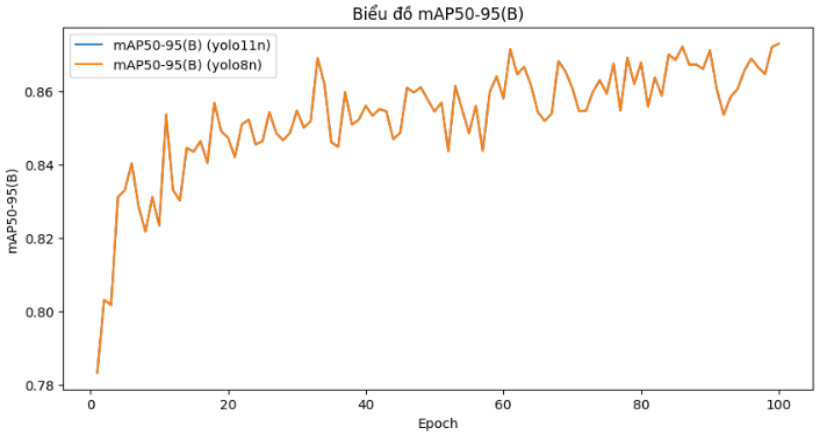


11. Bs_optimizer_epoch_lr: 4_AdamW_100_0.001

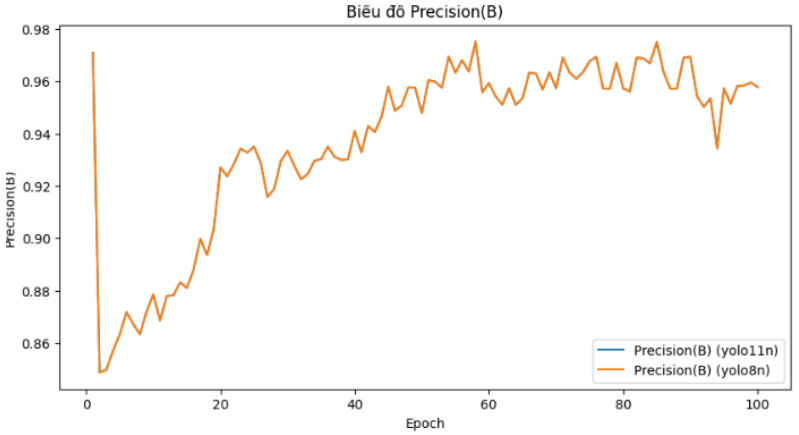


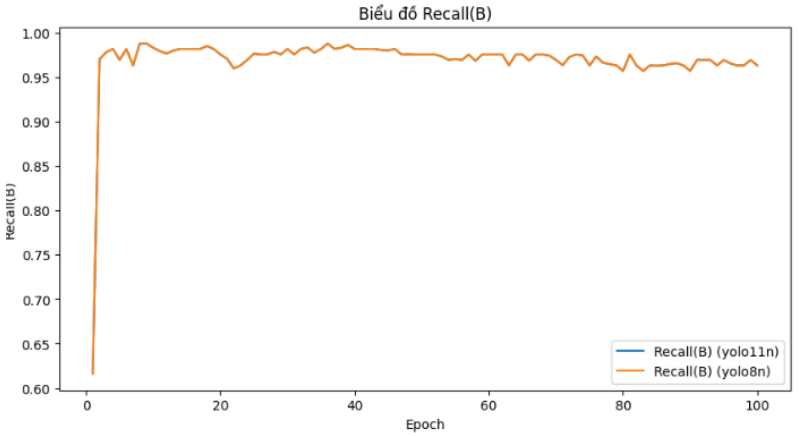


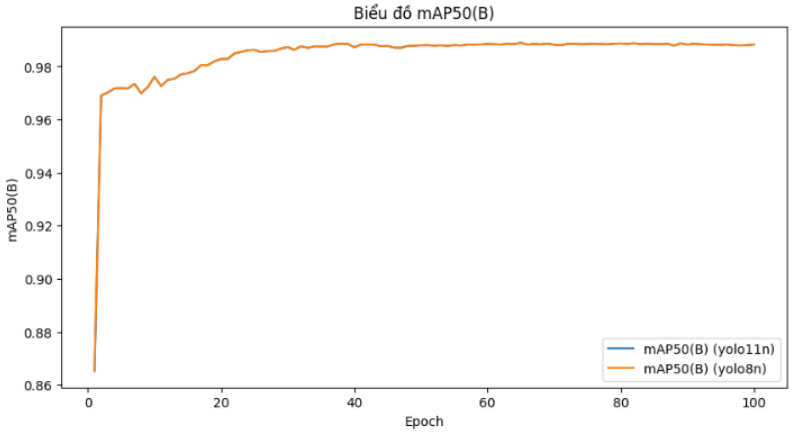


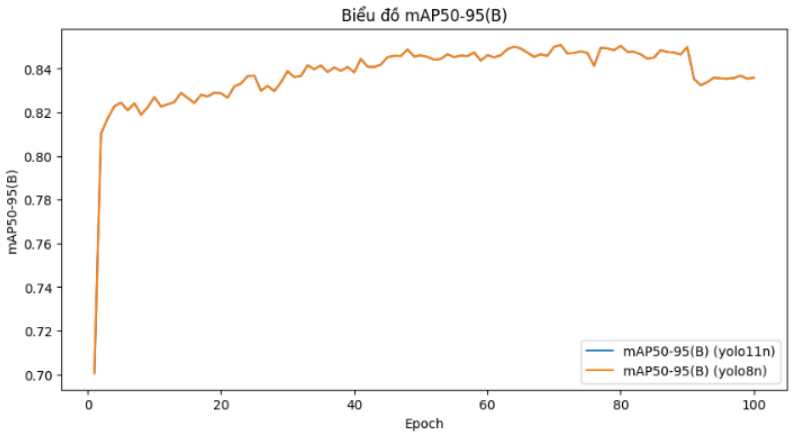


12. Bs_optimizer_epoch_lr: 4_AdamW_100_1e-5

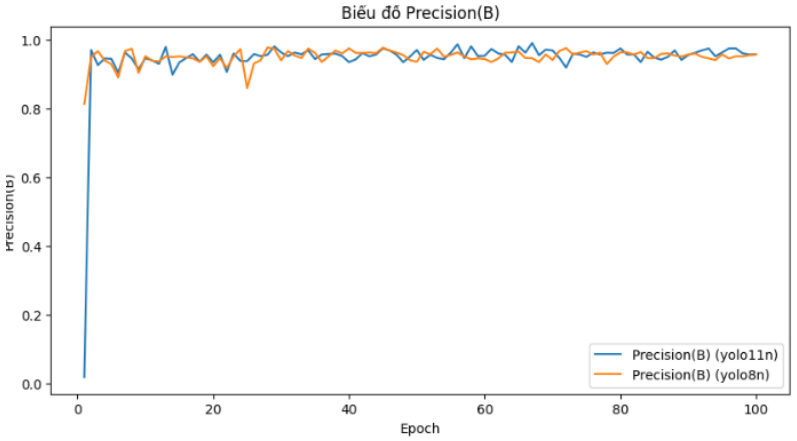


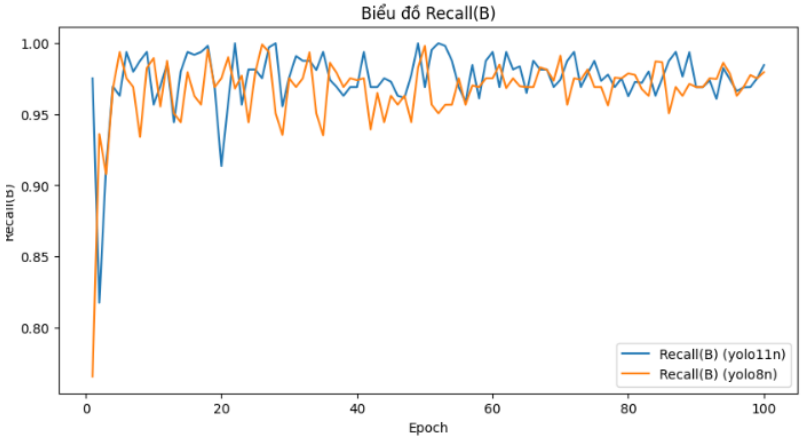


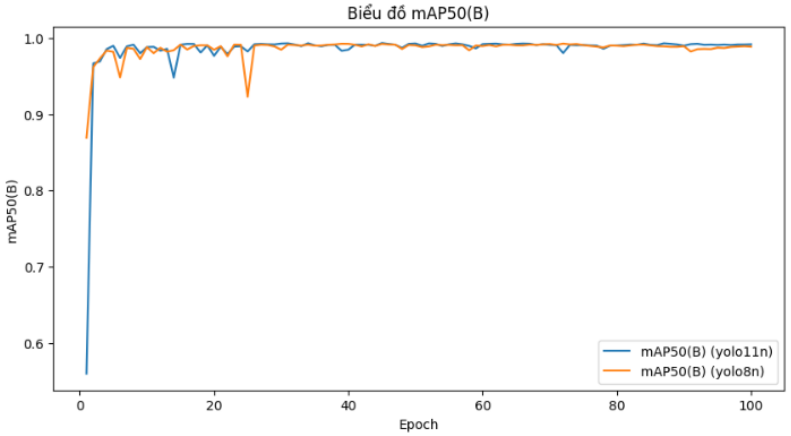


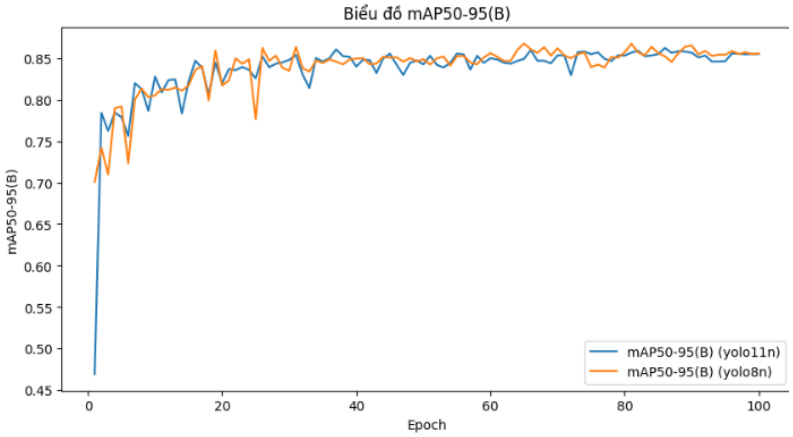


13. Bs_optimizer_epoch_lr: 8_SGD_100_0.01

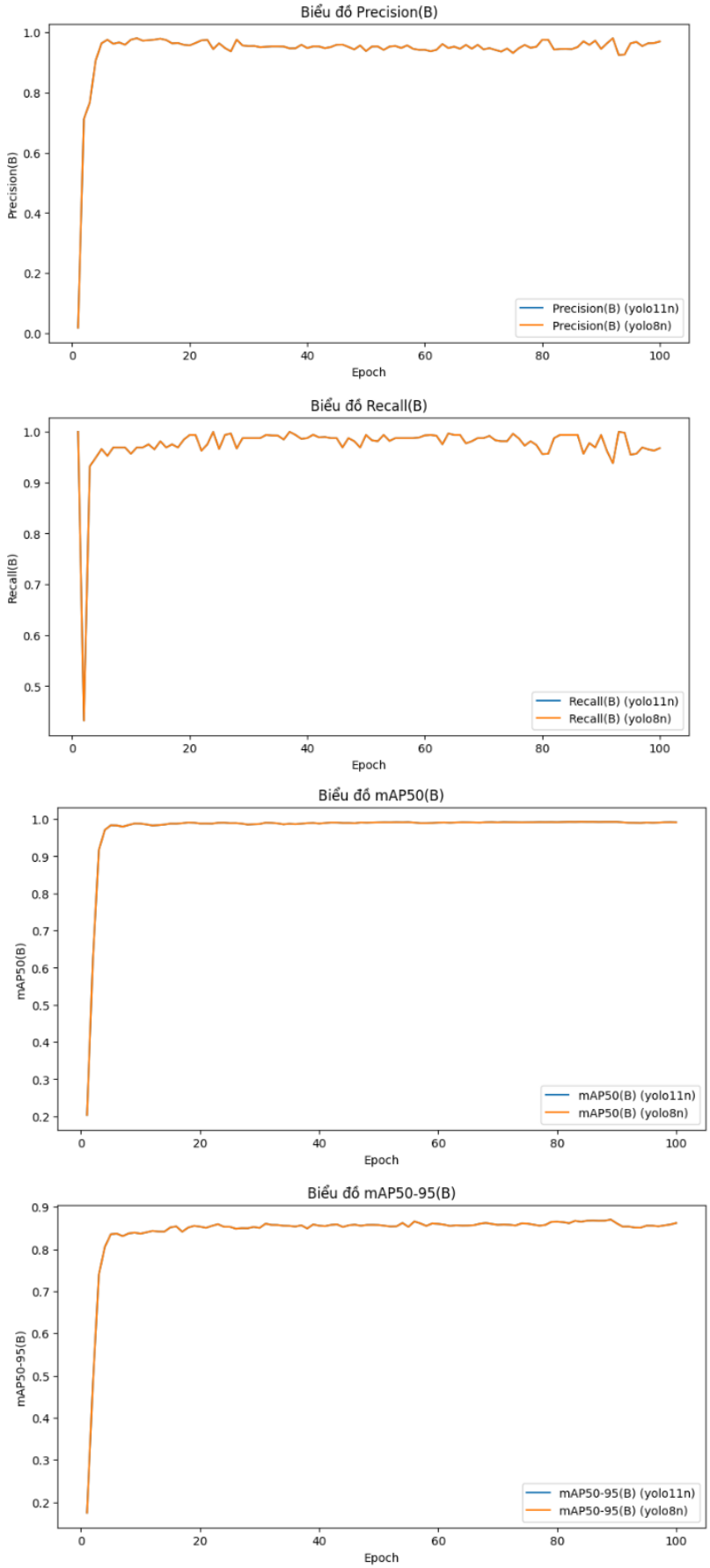




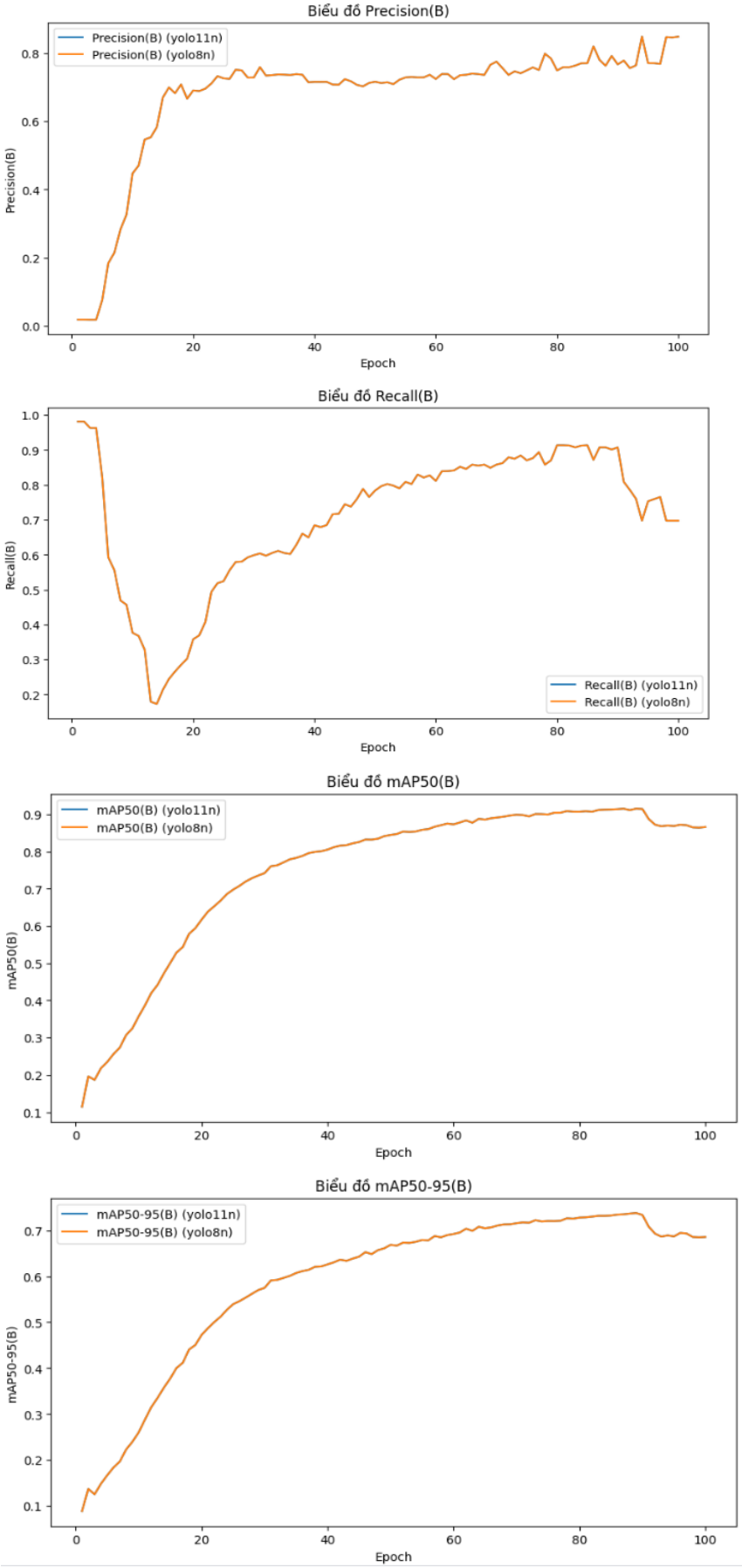




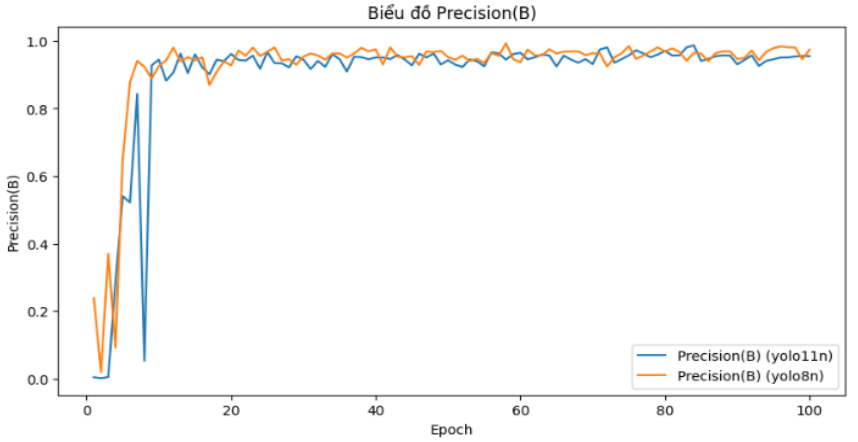
14. Bs_optimizer_epoch_lr: 8_SGD_100_0.001

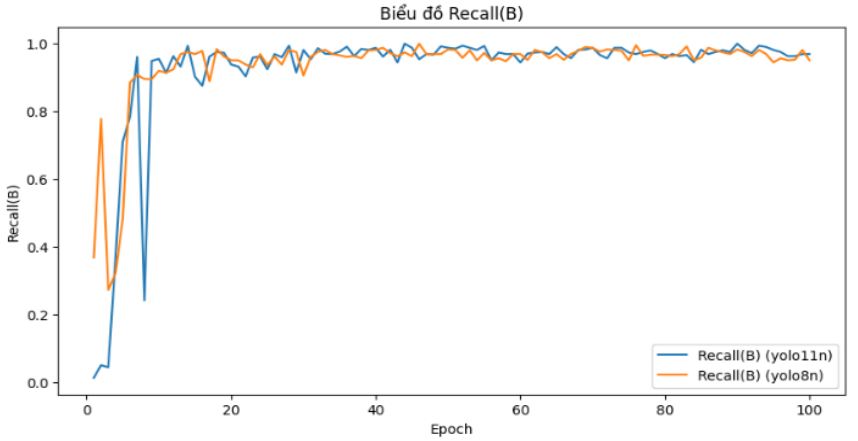


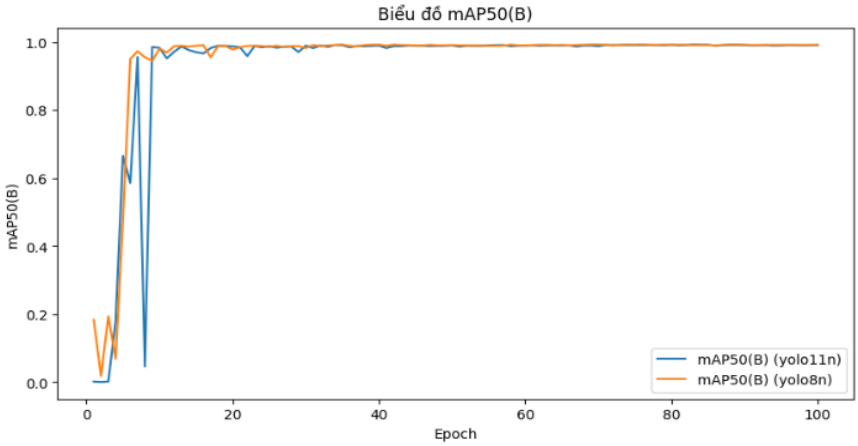
15. Bs_optimizer_epoch_lr: 8_SGD_100_1e-5

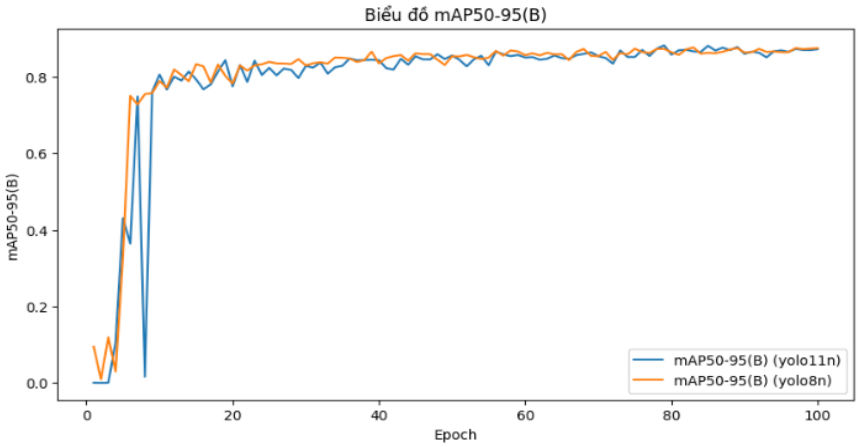


16. Bs_optimizer_epoch_lr: 8_AdamW_100_0.01

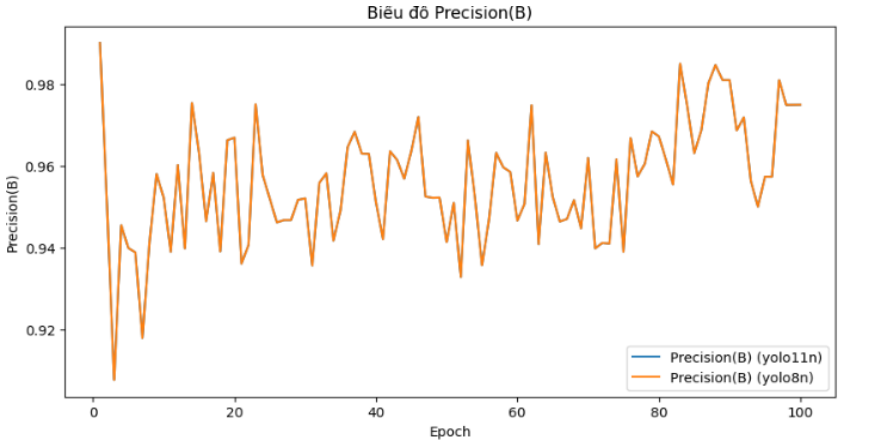


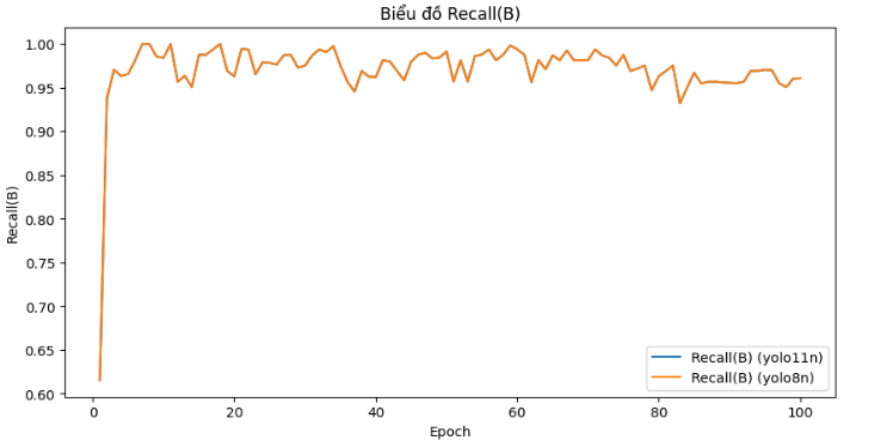


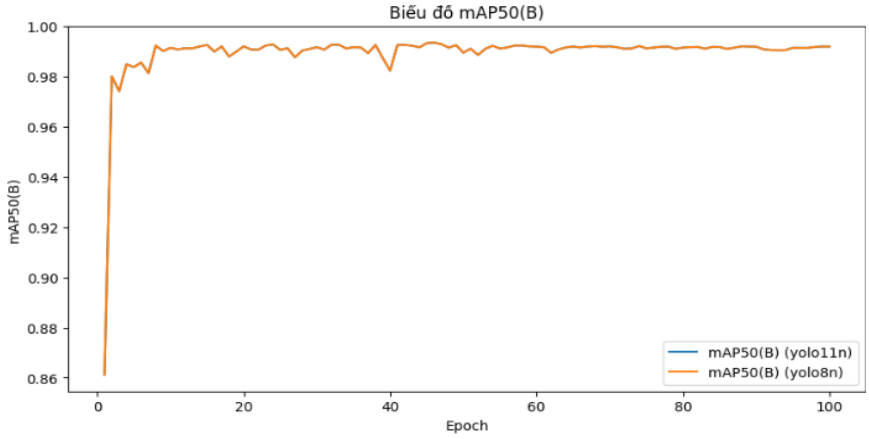


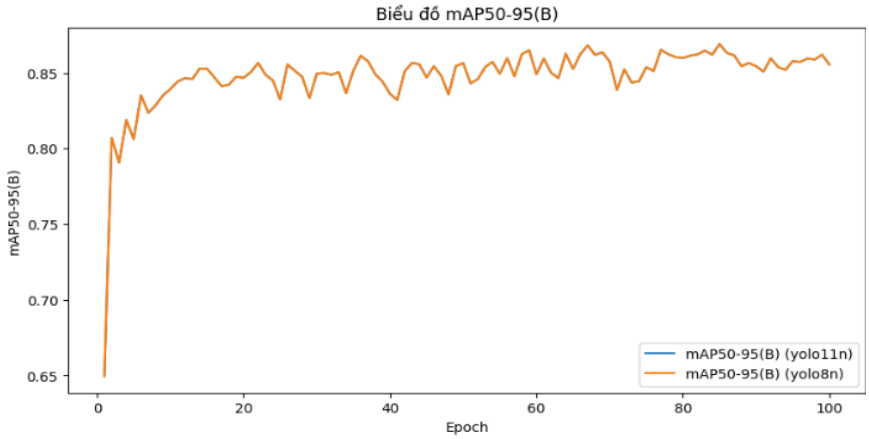


17. Bs_optimizer_epoch_lr: 8_AdamW_100_0.001

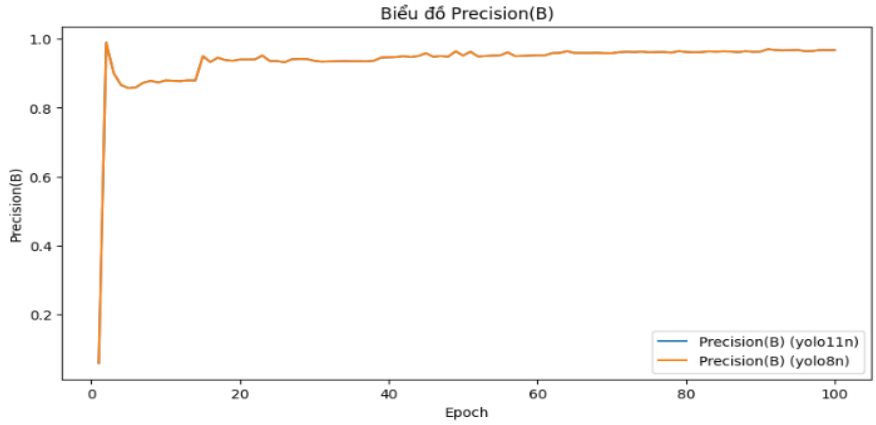


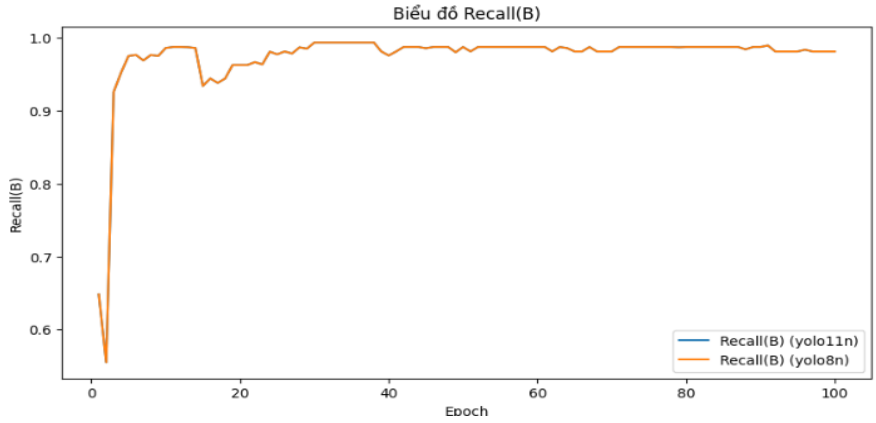


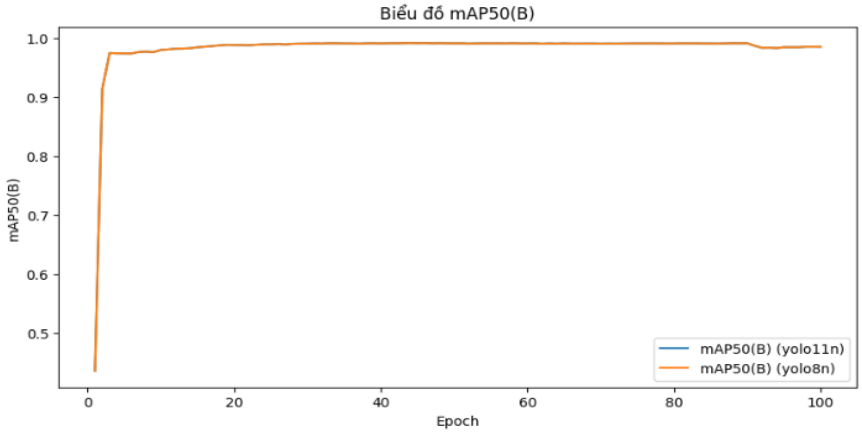


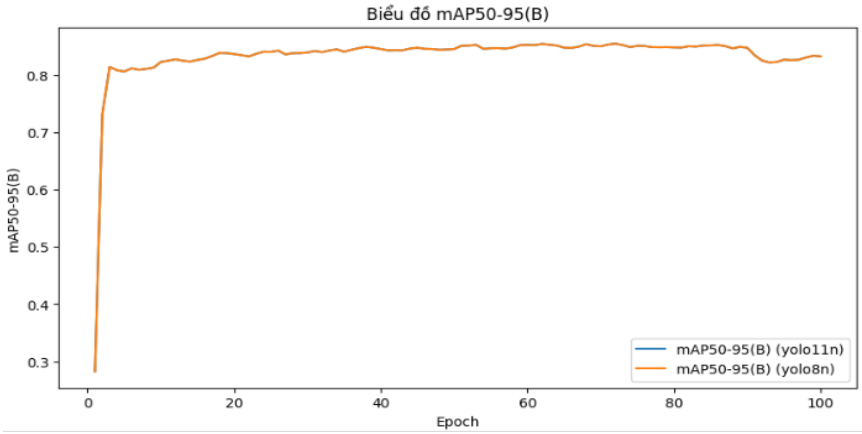


18. Bs_optimizer_epoch_lr: 8_AdamW_100_1e-5









3. Kết quả đánh giá

	FPS trung bình	AP@50-95	TG xử lí (GPU)	TG xử lí (CPU)
Yolov8n	12.76	0.752	30+-3 ms	63+-4 ms
Yolov11n	13.71	0.757	33+-2 ms	59+-5 ms

- Từ kết quả trên ta thấy được, đúng như sự cải tiến của YOLOv11, việc xác định các vật thể nhỏ và khó thì YOLOv11 làm tốt hơn (khi chỉ số mAP50-95 cao hơn 1 chút)
- Vì đánh giá trên GPU cho ảnh đơn, thì YOLOv8 làm tốt hơn, nhưng với điều kiện cấu hình thì YOLOv11 làm tốt hơn (khi tối ưu được việc tính toán trong điều kiện đơn nhân đơn luồng)

V. ĐÁNH GIÁ, NHẬN XÉT

1. Ưu điểm

- Với phương pháp giải quyết bài toán trên, có thể xử lí được với mức độ tạm ổn (mAP50-95 ~ 0.75) với lượng dữ liệu nhỏ, thời gian hạn hẹp.
- Tuning tìm ra các bộ parameters tốt cho các trường hợp trên.
- Đưa ra được các ưu, nhược điểm của từng phiên bản YOLO để có thể có sự lựa chọn hợp lí, cân bằng giữa các yếu tố để phù hợp với bài toán thực tế.
- Đánh giá cả 2 model trên các phần cứng khác nhau.

2. Nhược điểm

- Lượng dữ liệu nhỏ nên hạn chế trong việc tối ưu giải pháp.
- YOLOv11 là một mô hình vừa xuất hiện, có thể chưa tối ưu được toàn bộ các tham số và cho kết quả tốt nhất.
- Trong bài toán nêu trên, phạm vi xử lí còn chưa lớn (môi trường tối, thời tiết xấu, điều kiện ánh sáng, chất lượng ảnh,...).

NGUỒN THAM KHẢO

1. https://arxiv.org/html/2410.17725v1#S6
2. https://pchenlab.wordpress.com/2024/12/23/yolov8-vs-yolov11-same-dataset/
3. https://docs.ultralytics.com/vi/models/yolov8/
4. https://github.com/ultralytics/ultralytics
5. https://viso.ai/deep-learning/yolov8-guide/
6. https://debuggercafe.com/anchor-free-object-detection-inference-using-fcos-fully-connected-one-stage-object-detection/
7. https://www.miai.vn/2021/10/14/thu-tim-hieu-ve-map-do-luong-object-detection-model/
8. https://docs.ultralytics.com/vi/models/yolov8/#performance-metrics
9. https://medium.com/@nikhil-rao-20/yolov11-explained-next-level-object-detection-with-enhanced-speed-and-accuracy-2dbe2d376f71