ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN CUỐI KỲ

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

(YOLOV10: PHÁT HIỆN ĐỐI TƯỢNG THEO THỜI GIAN THỰC)

HỌC THỐNG KÊ

Giảng viên hướng dẫn:
Ngô Minh Nhựt
Lê Long Quốc

Thành phố Hồ Chí Minh, Ngày 4 tháng 9 năm 2024

Mục lục

1 Giới thiệu đồ án

1.1 Thông tin nhóm

MSSV	Họ và tên
21127590	Nguyễn Đức Tuấn Đạt
21127619	Phạm Gia Tuấn Khải
21127620	Trần Hoàng Khải
21127730	Hoàng Lê Cát Thanh

1.2 Giới thiệu về YOLO

- Phát hiện đối tượng theo thời gian thực là một bài toán quan trọng trong lĩnh vực thị giác máy tính, với mục tiêu nhận dạng và định vị các đối tượng trong hình ảnh hoặc video một cách nhanh chóng và chính xác. Những ứng dụng thực tiễn của bài toán này trải dài từ giám sát an ninh, phân tích giao thông, đến hỗ trợ cho các hệ thống tự động như robot tự hành và xe tự lái.

- Tuy nhiên bài toán này vẫn tồn tại một số thách thức trong quá trình giải quyết như

  - **Tốc độ và độ chính xác:** Để có thể hoạt động hiệu quả trong thời gian thực, mô hình cần xử lý hình ảnh hoặc video ở tốc độ cao mà vẫn đảm bảo độ chính xác trong việc nhận dạng và phân loại đối tượng.
  - **Dữ liệu thay đổi liên tục:** Các đối tượng trong hình ảnh/video có thể thay đổi về kích thước, hình dáng, vị trí, và bối cảnh, điều này đòi hỏi mô hình phải có khả năng xử lý tốt các tình huống đa dạng và phức tạp.
  - **Tài nguyên tính toán:** Việc xử lý một lượng lớn dữ liệu trong thời gian ngắn đòi hỏi hệ thống phải tối ưu về mặt tính toán, đặc biệt là khi triển khai trên các thiết bị phần cứng hạn chế.

- YOLO là một trong những mô hình tiên phong trong việc giải quyết bài toán phát hiện đối tượng theo thời gian thực. Khác với các phương pháp truyền thống, YOLO có khả năng xử lý toàn bộ hình ảnh chỉ trong một lần chạy mô hình, thay vì chia nhỏ hình ảnh thành các vùng khác nhau như khi sử dụng R-CNN. YOLO chia hình ảnh thành các ô lưới và dự đoán đồng thời vị trí, kích thước của các hộp chứa đối tượng và xác suất của các lớp đối tượng, từ đó xử lý và phát hiện đối tượng với tốc độ rất cao nhưng vẫn đảm bảo được độ chính xác của mô hình, phù hợp với yêu cầu của các ứng dụng thời gian thực.

- YOLOv10 là phiên bản mới nhất trong các mô hình YOLO, với những cải tiến đáng kể về hiệu suất và độ chính xác. Được xây dựng dựa trên những thành tựu của các phiên bản trước, YOLOv10 tích hợp các công nghệ học sâu tiên tiến để xử lý tốt hơn các tình huống phức tạp, như phát hiện các đối tượng nhỏ hoặc đối tượng trong môi trường có nhiều nhiễu.

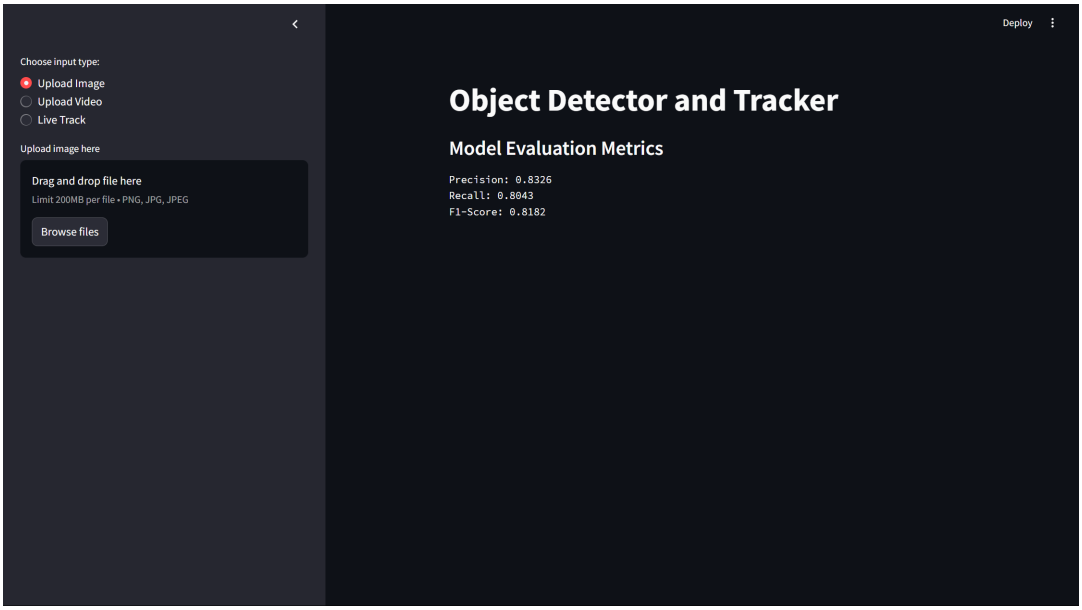
- YOLOv10 sử dụng một kiến trúc mạng nơ-ron tích chập (CNN) cải tiến, cho phép mô hình xử lý các đặc điểm hình ảnh phức tạp hơn với hiệu quả cao hơn. Mô hình này cũng sử dụng các kỹ thuật như FPN (Feature Pyramid Networks) và các chiến lược liên quan đến tăng cường

độ phân giải của các lớp đầu ra giúp nhận diện chính xác hơn những đối tượng có kích thước nhỏ mà trước đây có thể dễ bị bỏ sót nhưng vẫn duy trì hiệu suất cao về tốc độ xử lý.

- Từ những đặc điểm trên của YOLOv10, nhóm đã chọn sử dụng phiên bản này để triển khai cho dự án cuối kỳ.

2 Sử dụng mô hình YOLO có sẵn và xây dựng Web Demo

2.1 Giới thiệu về giao diện và chức năng của web demo.

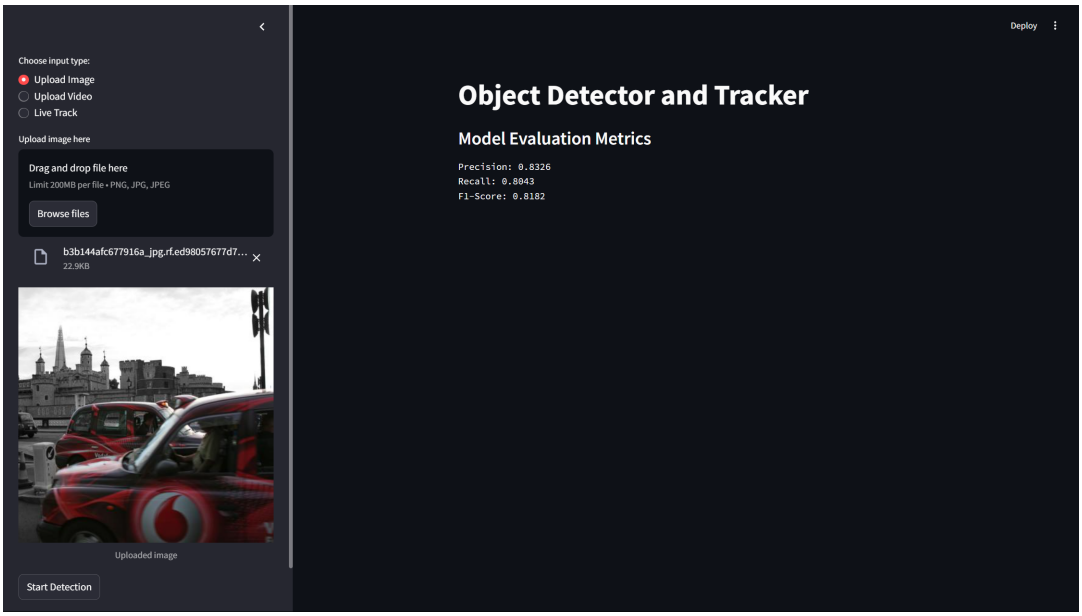


Hình 1: Giao diện Web Demo YOLOv10

- Giao diện ứng dụng được xây dựng bằng *Streamlit*, một framework mã nguồn mở của Python, cho phép phát triển giao diện web tương tác một cách nhanh chóng và đơn giản mà không cần nhiều kiến thức về front-end nhờ khả năng cung cấp các công cụ và tính năng giúp dễ dàng tạo ra các thành phần giao diện.

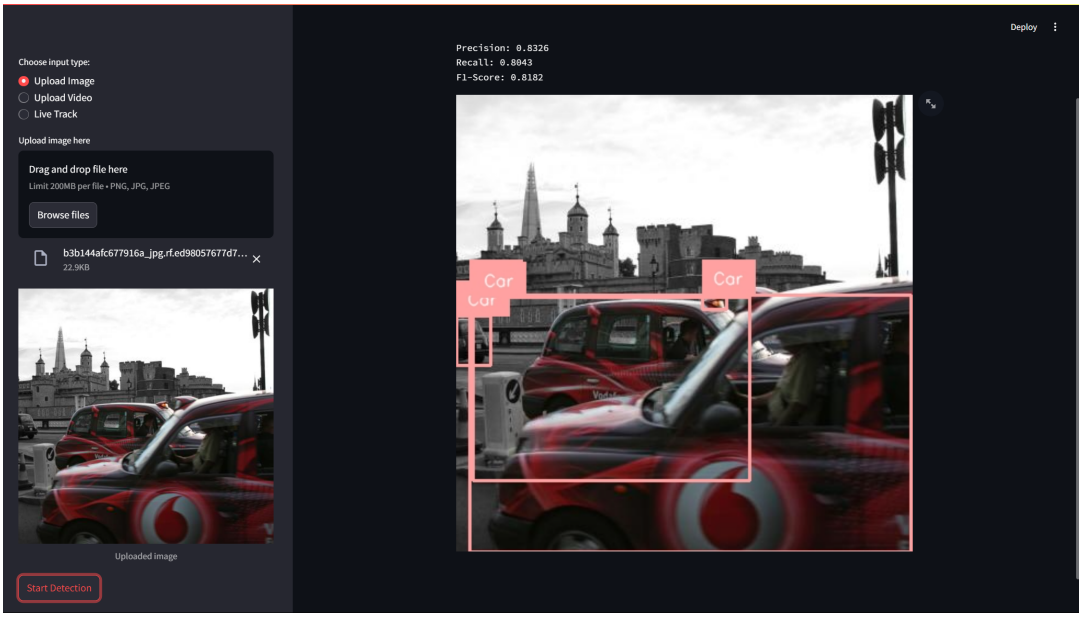
- Khi người dùng tải lên một hình ảnh đầu vào, nút **Start Detection** sẽ xuất hiện và khi nhấn vào nút này, quá trình phát hiện đối tượng trong hình ảnh sử dụng mô hình YOLO sẽ bắt đầu và kết quả trả về sẽ được hiển thị ngay trên giao diện, giúp người dùng dễ dàng quan sát và đánh giá.

- Vì YOLOv10 là mô hình nhận diện vật thể trong khung hình dưới dạng video theo thời gian thực, vậy nên nhóm có cài đặt thêm 2 lựa chọn cho Web Demo khi ứng dụng YOLOv10, đó là lựa chọn Upload Video cho phép nhận diện vật thể trong video và Live Track cho phép nhận diện vật thể trực tiếp thông qua thiết bị đầu vào camera.



Hình 2: Giao diện sau khi upload ảnh

2.2 Thực nghiệm trên Web Demo



Hình 3: Kết quả thực nghiệm trên mô hình có sẵn

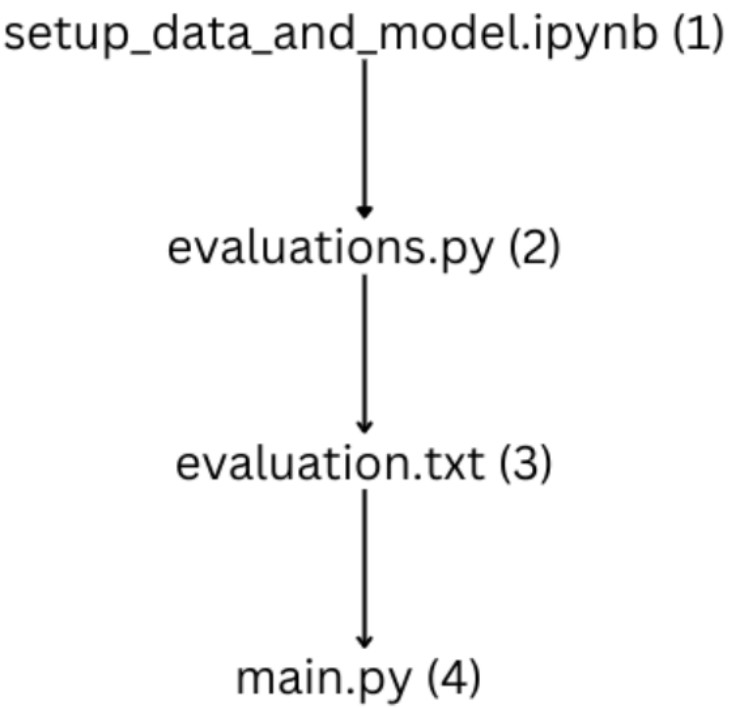
- Khi người dùng nhấn vào nút **Start Detection**, mô hình YOLO sẽ thực hiện tìm kiếm và phát hiện các đối tượng trong hình ảnh. Quá trình này bao gồm việc phân tích hình ảnh để xác định vị trí của các đối tượng, sau đó vẽ các bounding boxes xung quanh các đối tượng đã được phát hiện.

- Ảnh kết quả trả về sẽ hiển thị các đối tượng được đánh dấu kèm theo tên của đối tượng mà

mô hình dự đoán được, cho biết mức độ chính xác của mô hình khi xác định tên của đối tượng, giúp người dùng hiểu rõ hơn về mức độ tin cậy của từng kết quả phát hiện.

3 Hướng dẫn chạy chương trình

3.1 Thứ tự cài đặt

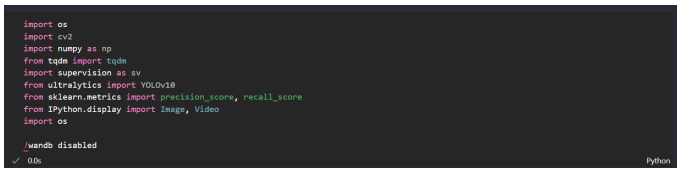


Hình 4: flow cài đặt

Đầu tiên là cần chạy file **setup_data_and_model.ipynb** để set up môi trường, tạo **file data.yaml** và huấn luyện mô hình. Sau đó chạy file **Evaluation.py** để lấy metrics đánh giá của mô hình sau khi huấn luyện, metrics lưu vào **evaluatións.txt**, sau đó mới chạy file **main.py** để ứng dụng mô hình

3.2 Từng bước cài đặt

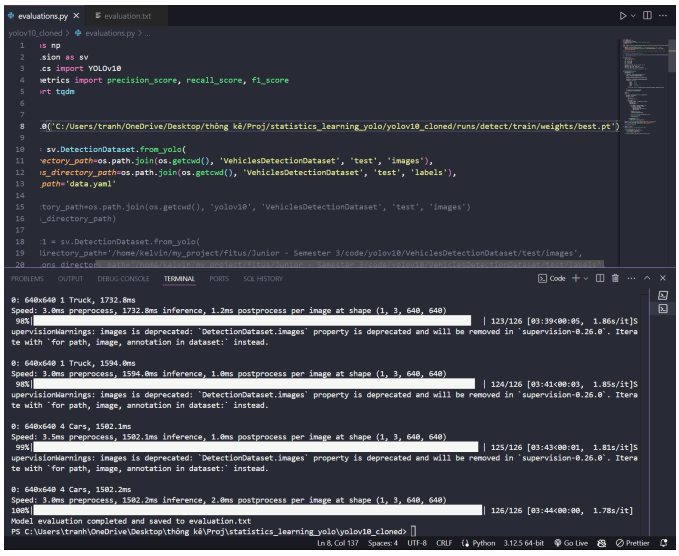
B1: Mở file **setup_data_and_model.ipynb**



Hình 5: chạy cell import library

Hình 6: chạy cell create data.yaml

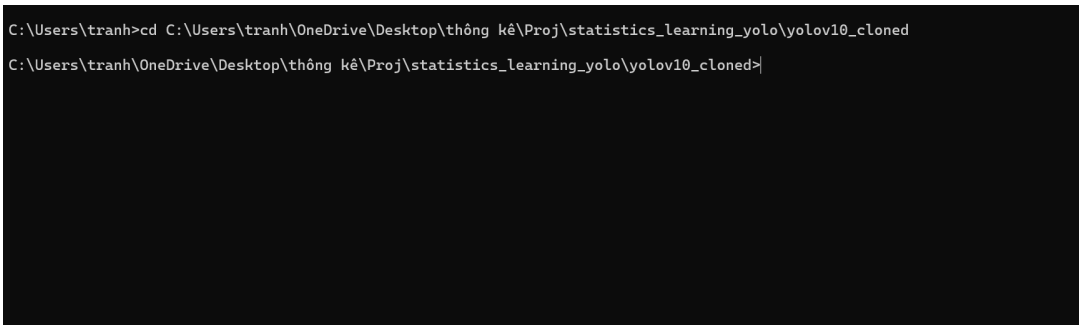
B2: Chạy file **evaluations.py** và metrics được ghi vào file **evaluation.txt**



Hình 7: Chạy và đợi chương trình evaluations

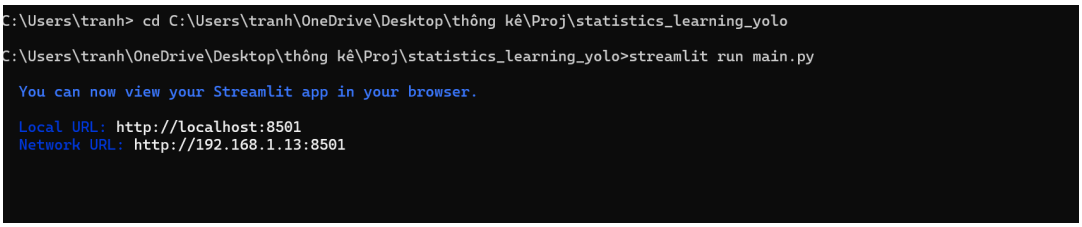
Hình 8: Precision,Recall, F1-Score

B3: Mở **commad prompt** và di chuyển đến thư mục theo đúng đường dẫn 'statistics_learning_yolo/yolov10_

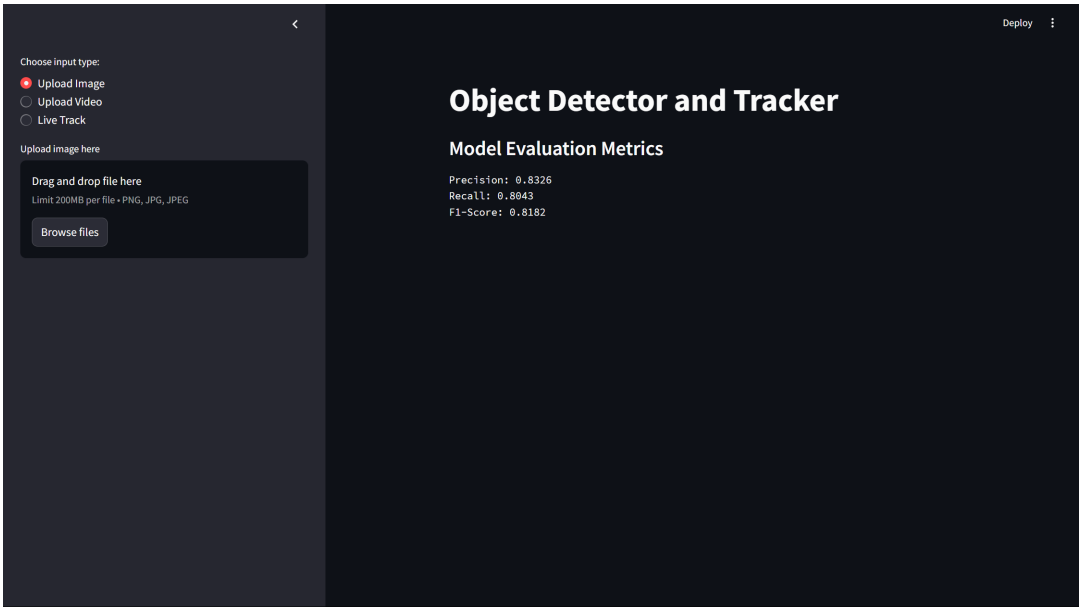


Hình 9: cmd đã thay đổi thư mục

B4: Nhập lệnh **streamlit run main.py** và nhấn **Enter**.



Hình 10: khởi tạo web interface



Hình 11: Tự nhảy vào web interface

4 Huấn luyện mô hình YOLO trên tập dữ liệu

4.1 Quá trình huấn luyện mô hình YOLOv10 diễn ra như thế nào?

Quá trình huấn luyện mô hình YOLO được thực hiện qua các bước:

- **Chuẩn bị dữ liệu:** thu thập dữ liệu sử dụng cho quá trình huấn luyện bao gồm các hình ảnh chứa các đối tượng cần nhận diện, cùng với các nhãn tương ứng được gán cho từng đối tượng trong hình và tiền xử lý để phù hợp với định dạng yêu cầu của phiên bản YOLO sử dụng. Nhãn của ảnh thường được lưu trữ dưới định dạng *.txt* theo format YOLO (label, x_center, y_center, width, height).

Hình 12: Cấu trúc folder chứa dataset

- **Tạo file mô tả:** quá trình huán luyện mô hình của YOLOv10 sử dụng một file mô tả *data.yaml* bao gồm thông tin về đường dẫn đến các tập train, test, validation, số lượng lớp và tên các lớp có trong tập dữ liệu.

```
test: /kaggle/input/vehicledetection/VehiclesDetectionDataset/test/images
train: /kaggle/input/vehicledetection/VehiclesDetectionDataset/train/images
val: /kaggle/input/vehicledetection/VehiclesDetectionDataset/valid/images

nc: 5  # Number of classes
names: ['Ambulance', 'Bus', 'Car', 'Motorcycle', 'Truck']
```

Hình 13: File mô tả dữ liệu sử dụng

- **Cấu hình mô hình:** lựa chọn kiến trúc mô hình phù hợp và thiết lập các tham số cho quá trình huấn luyện như số epoch, kích thước batch,...
- **Huấn luyện mô hình:** mô hình được huấn luyện trên tập dữ liệu đã chuẩn bị, với các trọng số của mô hình được cập nhật liên tục qua mỗi epoch để tối ưu hóa độ chính xác nhận diện đối tượng. Sau mỗi epoch, mô hình được kiểm tra và đánh giá tra các chỉ số như Precision, mAP.

```
!yolo task=detect mode=train epochs=100 batch=16 plots=True model=yolov10l.pt data=data.yaml
```

Hình 14: Huấn luyện mô hình

4.2 Giới thiệu tập dữ liệu được dùng để huấn luyện

- Chủ đề nhận diện của nhóm sẽ là "Nhận diện phương tiện giao thông" bao gồm 5 phương tiện chính như là: xe hơi(Car), xe tải (Truck), xe máy (Motorcycle), xe cứu thương (Ambulance) và xe buýt (Bus).
- Tập dữ liêu được sử dụng để huấn luyện mô hình có tên là VehiclesDetectionDataset. Trong tập dữ liệu này, các tập Train/Test/Validation đã được tác giả chia ra sẵn ra thành 2 thư mục là hình ảnh (images) và nhãn (labels), với lần lượt từng set có số lượng ảnh là 878, 126 và 250. Tương ứng với tỉ lệ của Train/Test/Validation là 75/10/15.
- Sau quá trình kiểm tra và thử nghiệm thì nhóm nhận thấy đây là bộ dữ liệu đưa ra hiệu quả nhận diện khá ổn, đủ để mô hình YOLOv10 của nhóm hoạt động tốt.

4.3 Phiên bản YOLOv10l có gì tối ưu so với các phiên bản khác trong chủ đề mà nhóm chọn?

- **Cân bằng giữa độ chính xác và tốc độ xử lý:** YOLOv10l cung cấp độ chính xác cao hơn so với các phiên bản nhẹ hơn như YOLOv10-N và S, nhưng vẫn giữ được tốc độ xử lý nhanh, phù hợp cho các ứng dụng yêu cầu nhận diện chính xác mà không quá đòi hỏi về tài nguyên.

- **Khả năng phát hiện đối tượng tốt:** YOLOv10l có số lượng lớp (layers) và tham số lớn hơn so với các phiên bản nhẹ hơn như YOLOv10-N và YOLOv10-S. Điều này giúp mô hình có khả năng học và phát hiện tốt hơn đối với các đối tượng nhỏ hoặc các đối tượng trong những tình huống phức tạp.
- **Khả năng mở rộng và ứng dụng rộng rãi:** YOLOv10l có tính linh hoạt cao, giúp triển khai trên nhiều ứng dụng yêu cầu tài nguyên hệ thống khác nhau.

4.4 Môi trường dùng để huấn luyện mô hình của nhóm là gì?

- Trong quá trình huấn luyện thì nhóm có sử dụng kaggle với GPU T4 x2 để huấn luyện mô hình YOLOv10. Việc này giúp nhóm tận dụng được tối đa sức mạnh GPU ảo trên Kaggle.

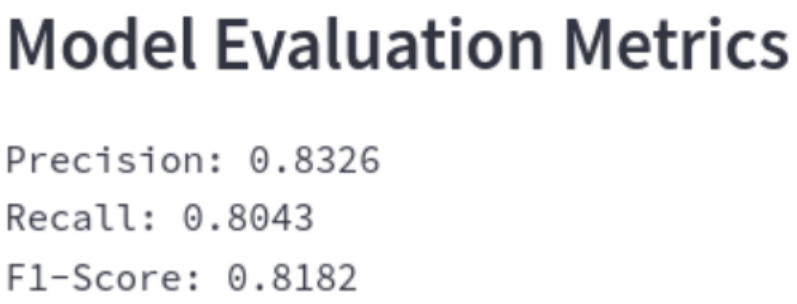
4.5 Mô hình có thỏa mãn yêu cầu?

Mô hình chỉ được phép tải một lần và được sử dụng cho tất cả các phân loại. Chỉ khi mô hình cần được cập nhật, hệ thống mới tải lại mô hình. Trong trường hợp vi phạm quy tắc này, bạn sẽ bị trừ 60% điểm cho yêu cầu thứ 1 và 40% điểm cho yêu cầu thứ 2.
Với luồng làm việc mà trước đó nhóm có đề cập trong việc khởi động Web Demo thì rõ ràng mô hình sau khi huấn luyện sẽ được lưu vào trong đường dẫn 'runs/detect/train/best.pt'. Và file .pt này sẽ lưu trữ mô hình mới nhất sau mỗi lần nhóm huấn luyện, và khi cần deploy Web Demo thì nhóm mới gọi lại mô hình này để có thể ứng dụng trọng việc nhận diện vật thể.

4.6 Đánh giá

4.6.1 Các metrics được sử dụng bao gồm những gì?

- Trong phần đánh giá dữ liệu, nhóm có sử dụng các metrics cơ bản để kiểm tra độ hiệu quả của mô hình sau khi được xử lí, bao gồm: Recall, Precision và F1-score.



Hình 15: Thông số đánh giá của mô hình sau khi huấn luyện

- Precison là thước đó xác định mức độ chính xác trong các dự đoán "đúng" mà mô hình thực hiện. Với giá trị là 0.8326 cho thấy rằng trong số các đối tượng mà mô hình dự đoán là đúng, có khoảng 83.26% là chính xác. Điều này thể hiện mô hình có khả năng phân loại đối tượng đúng khá tốt, nhưng vẫn có khoảng 16.74% là các dự đoán sai (False Positives).

- Recall đạt 0.8043, có nghĩa là mô hình có khả năng phát hiện được khoảng 80.43% các đối tượng thật sự tồn tại. Đây là một chỉ số khá tốt, nhưng nó cho thấy vẫn còn một số đối tượng bị bỏ sót (False Negatives), có thể do mô hình chưa nhạy cảm đủ với tất cả các đặc điểm của các đối tượng.

- F1-Score đạt 0.8182, đây là một chỉ số cân bằng giữa Precision và Recall. F1-Score cho thấy mô hình đang duy trì một sự cân đối tốt giữa việc phát hiện đúng đối tượng và hạn chế các dự đoán sai.
  Kết luận: Các chỉ số này cho thấy mô hình YOLOv10 có hiệu suất tổng thể khá tốt, với khả năng phát hiện đối tượng cao và độ chính xác trong các dự đoán dương cũng đáng kể. Tuy nhiên, nếu mô hình gặp phải nhiều trường hợp bỏ sót đối tượng quan trọng (Recall thấp hơn), chúng ta cần cân nhắc điều chỉnh lại mô hình hoặc dữ liệu huấn luyện để cải thiện chỉ số Recall mà không ảnh hưởng quá nhiều đến Precision. Đây cũng chính là điểm yếu của mô hình của nhóm, và giải pháp nhóm tạm thời tìm ra đó chính là nạp thêm dữ liệu đầu vào để mô hình có thể nhận diện nhiều hơn và đúng hơn các vật thể mà nhóm mong muốn.

4.6.2 Mỗi metric trên mỗi set trên có kết quả như thế nào?

Đối với tập huấn luyện:

- Precision: 0.8515
- Recall: 0.8388
- F1-score: 0.8451

Đối với Validation:

- Precision: 0.8212
- Recall: 0.8133
- F1-score: 0.8172

Kết luận: Nhìn chung, thông số trên Train Set và Validation Set có nhỉnh hơn so với Test Set, nhưng chúng lại không có sự chênh lệch về giá trị quá lớn.

4.6.3 Mô hình YOLOv10 ứng dụng trên Web Demo

- Nhìn chung, mô hình của nhóm vẫn đáp ứng được đúng yêu cầu đồ án do giảng viên đưa ra. Input khi truyền vào sau khi được mô hình xử lí sẽ đưa ra output với kết quả nhận diện khá tốt.

- Tuy nhiên, với việc thông số đang không quá cao, ở dưới mức 0.9 với mọi thông số, thì kết quả đầu ra sẽ có những lúc không nhận diện đúng như chúng ta mong muốn. Đặc biệt với những input đầu vào với chất lượng không tốt, hoặc có nhiều vật thể gây nhiễu khiến mô hình khó nhận diện được chính xác.

5 Tài liệu tham khảo

[1] Tài liệu môn học