

BÁO CÁO NỘI DUNG ĐÃ TÌM HIỂU KHÓA INTERMEDIATE

Đề tài: Nhận diện biển số xe

Họ tên: Phạm Việt Hùng

Mã Gen: 25420066

Hà Nội, ngày 12 tháng 3 năm 2025

Mục lục

Giới thiệu	3
1 Tổng quan lý thuyết	4
1.1 Giới thiệu về YOLO (You Only Look Once)	4
1.1.1 Phiên bản sử dụng	4
1.1.2 Cách hoạt động của YOLO	4
1.1.3 Tại sao sử dụng YOLO thay vì các phương pháp khác	5
2 Phương pháp và Công nghệ	6
2.1 Mô hình YOLOv8	6
2.1.1 Các thông số cấu hình	6
2.2 Công cụ và môi trường	7
2.2.1 Google Colab	7
2.2.2 VSCode (Visual Studio Code)	7
2.2.3 Công cụ và Thư viện	7
3 Quá trình thực hiện	8
3.1 Chuẩn bị dữ liệu	8
3.1.1 Thu thập dữ liệu	8
3.1.2 Xử lý và chuẩn bị dữ liệu	8
3.1.3 Tạo và gán annotation	8
3.1.4 Phân chia dữ liệu	9
3.2 Huấn luyện mô hình	9
4 Quá Trình Huấn Luyện Mô Hình	10
4.1 Giới thiệu	10
4.2 Thu Thập Dữ Liệu	10
4.2.1 Dữ Liệu Biển Số Xe	10
4.2.2 Dữ Liệu Ký Tự	10
4.3 Chuẩn Bị Dữ Liệu	12
4.4 Huấn Luyện Mô Hình	13

5	Phân Tích Kết Quả Huấn Luyện Mô Hình	16
5.1	Các Chỉ Số Đánh Giá	16
5.2	Đánh Giá Kết Quả Huấn Luyện	17
5.2.1	Phân Tích Mất Mát	17
5.2.2	Hiệu Suất Mô Hình	17
5.3	Hình Ảnh Minh Họa	17
5.4	Kết Quả Đánh Giá Mô Hình	18
5.4.1	Kết Quả Phát Hiện Biến Số	18
5.4.2	Điểm Yếu và Các Tình Huống Khó	20
5.5	Kết Luận và Hướng Đi Tiếp Theo	20
5.5.1	Kết Luận	20
5.5.2	Hướng Đi Tiếp Theo	20

Giới thiệu

Trong kỷ nguyên số hiện nay, trí tuệ nhân tạo (AI) và học sâu (deep learning) đã và đang trở thành những công nghệ tiên phong trong việc cải thiện và tự động hóa nhiều lĩnh vực khác nhau. Các ứng dụng của AI không chỉ giúp tối ưu hóa quy trình làm việc mà còn mở ra những khả năng mới trong việc phân tích và xử lý dữ liệu. Một trong những lĩnh vực đang được chú trọng là nhận diện đối tượng (Object Detection), và một trong những ứng dụng quan trọng của nó là nhận diện biển số xe.

Nhận diện biển số xe là một bài toán phức tạp và đầy thách thức, đặc biệt trong các tình huống thực tế như điều kiện ánh sáng yếu, góc nhìn không đồng nhất, và các yếu tố gây nhiễu khác. Việc tự động nhận diện và đọc biển số xe có ứng dụng rộng rãi trong nhiều lĩnh vực, bao gồm quản lý giao thông, giám sát an ninh, và phát triển các hệ thống tự động hóa. Đây là một phần quan trọng trong việc xây dựng các hệ thống an ninh thông minh và các giải pháp quản lý giao thông hiệu quả.

YOLO (You Only Look Once) là một trong những mô hình tiên tiến nhất trong lĩnh vực nhận diện đối tượng. Được biết đến với khả năng phân tích và xử lý hình ảnh nhanh chóng và chính xác, YOLO đã trở thành công cụ phổ biến trong các ứng dụng thời gian thực. Phiên bản mới nhất, YOLOv8, mang lại những cải tiến vượt bậc về độ chính xác và hiệu suất. YOLOv8 không chỉ cải thiện khả năng nhận diện đối tượng mà còn tối ưu hóa quy trình học máy, giúp mô hình hoạt động hiệu quả hơn trong các điều kiện thực tế.

Trong kỳ thực tập này, tôi đã thực hiện nghiên cứu và triển khai mô hình YOLOv8 nhằm giải quyết bài toán nhận diện biển số xe. Quá trình thực hiện bao gồm nhiều giai đoạn quan trọng như chuẩn bị dữ liệu, huấn luyện mô hình, và tích hợp mô hình vào hệ thống ứng dụng thực tế. Báo cáo này sẽ trình bày chi tiết từng bước trong quá trình thực hiện, từ việc thu thập và xử lý dữ liệu đến việc đánh giá hiệu suất của mô hình. Mục tiêu của báo cáo không chỉ là trình bày các kết quả đạt được mà còn phân tích các thách thức, bài học kinh nghiệm, và những hướng phát triển tiếp theo cho dự án.

Chương 1

Tổng quan lý thuyết

1.1 Giới thiệu về YOLO (You Only Look Once)

YOLO (You Only Look Once) là một trong những thuật toán nhận diện đối tượng tiên tiến và được sử dụng rộng rãi nhất trong lĩnh vực thị giác máy tính (Computer Vision). Được phát triển bởi Joseph Redmon và các cộng sự vào năm 2015, YOLO đã tạo ra bước ngoặt trong việc nhận diện đối tượng nhờ khả năng xử lý tốc độ cao và độ chính xác ấn tượng.

1.1.1 Phiên bản sử dụng

YOLOv(2023): Là phiên bản mới nhất, YOLOv8 tích hợp nhiều cải tiến từ các phiên bản trước, đồng thời cung cấp một framework mạnh mẽ cho việc huấn luyện và triển khai mô hình. YOLOv8 tối ưu hóa hơn về kích thước mô hình, tốc độ suy luận và khả năng tổng quát hóa trên các bộ dữ liệu khác nhau. Điều này làm cho YOLOv8 trở thành một lựa chọn hàng đầu trong các ứng dụng thời gian thực, đặc biệt là khi nhận diện xe hơi trong video hoặc hình ảnh.

1.1.2 Cách hoạt động của YOLO

YOLO hoạt động dựa trên nguyên tắc chia hình ảnh đầu vào thành một lưới $S \times S$. Mỗi ô trong lưới này sẽ dự đoán các bounding boxes và xác suất lớp đối tượng mà nó chứa. Các bounding boxes này bao gồm thông tin về vị trí trung tâm, chiều rộng, chiều cao, và độ tin cậy (confidence score) của đối tượng trong ô đó.

1.1.3 Tại sao sử dụng YOLO thay vì các phương pháp khác

Có nhiều lý do khiến YOLO trở thành lựa chọn ưu tiên trong các bài toán nhận diện đối tượng:

- **Tốc độ cao:** Khả năng xử lý hình ảnh chỉ trong một bước duy nhất giúp YOLO có thể đạt được tốc độ suy luận rất cao, phù hợp với các ứng dụng yêu cầu thời gian thực như giám sát giao thông, tự động hóa xe, và nhận diện xe hơi.
- **Hiệu suất tốt:** Dù tốc độ nhanh, YOLO vẫn duy trì được độ chính xác cao trong việc nhận diện các đối tượng, đặc biệt với các phiên bản cải tiến như YOLOv8.
- **Khả năng tổng quát hóa:** YOLO hoạt động tốt trên nhiều loại đối tượng khác nhau và có thể dễ dàng điều chỉnh để phù hợp với nhiều tình huống cụ thể.
- **Dễ triển khai và sử dụng:** Với sự hỗ trợ của nhiều framework như Darknet, PyTorch, TensorFlow, và khả năng tích hợp dễ dàng với các mô hình khác thông qua định dạng ONNX.

Chương 2

Phương pháp và Công nghệ

2.1 Mô hình YOLOv8

YOLOv8 là phiên bản mới nhất của thuật toán YOLO, tiếp tục cải thiện về độ chính xác và hiệu suất so với các phiên bản trước đó. YOLOv8 tích hợp các cải tiến về cấu trúc mạng, các kỹ thuật huấn luyện và các phương pháp tối ưu hóa, giúp nâng cao khả năng nhận diện đối tượng trong thời gian thực.

2.1.1 Các thông số cấu hình

- **Kích thước đầu vào:** YOLOv8 thường yêu cầu hình ảnh đầu vào có kích thước cố định, chẳng hạn như 640x640 pixels.
- **Số lớp và kích thước mạng:** YOLOv8 sử dụng kiến trúc mạng tối ưu hóa, có thể bao gồm nhiều lớp nơ-ron và các kỹ thuật mới như CSPNet, PANet, và các lớp chú ý (attention layers).
- **Anchor Boxes:** Được tinh chỉnh để phù hợp với kích thước đối tượng trong bộ dữ liệu, cải thiện khả năng nhận diện cho các đối tượng có kích thước khác nhau.
- **Các siêu tham số huấn luyện:** Bao gồm tỷ lệ học (learning rate), số epoch, kích thước batch, và các tham số liên quan đến việc tối ưu hóa mô hình.

2.2 Công cụ và môi trường

2.2.1 Google Colab

Google Colab là một nền tảng cung cấp môi trường phát triển miễn phí với GPU/TPU hỗ trợ. Nó được sử dụng để huấn luyện mô hình YOLOv8, cho phép triển khai và chạy các tập lệnh Python mà không cần cấu hình phần cứng địa phương. Các lợi ích của Google Colab bao gồm khả năng truy cập vào tài nguyên tính toán mạnh mẽ và tích hợp dễ dàng với Google Drive để lưu trữ dữ liệu và mô hình.

2.2.2 VSCode (Visual Studio Code)

VSCode là một môi trường phát triển tích hợp (IDE) mạnh mẽ, được sử dụng để viết và quản lý mã nguồn của dự án. Với các tiện ích mở rộng và tích hợp, VSCode hỗ trợ việc phát triển mã Python, gỡ lỗi, và quản lý các tệp dự án một cách hiệu quả. Các plugin và tiện ích mở rộng như Python Extension, Jupyter, và Git Integration giúp tăng cường quy trình phát triển và kiểm soát phiên bản.

2.2.3 Công cụ và Thư viện

- **Python:** Ngôn ngữ lập trình chính được sử dụng để viết các tập lệnh huấn luyện và triển khai mô hình.
- **PyTorch/TensorFlow:** Các thư viện học máy chính được sử dụng để xây dựng và huấn luyện mô hình YOLOv8. YOLOv8 có thể được triển khai trên PyTorch để tận dụng khả năng tính toán hiệu quả.
- **Flask:** Được sử dụng để xây dựng ứng dụng web cho việc triển khai mô hình nhận diện đối tượng. Flask giúp tạo ra một API để nhận và xử lý hình ảnh từ người dùng, đồng thời trả về kết quả nhận diện từ mô hình.
- **Ultralytics:** Thư viện Ultralytics tương thích tốt với các framework học sâu như PyTorch và TensorFlow, và dễ dàng tích hợp vào các môi trường phát triển như Google Colab và VSCode. Điều này giúp đơn giản hóa quy trình phát triển và triển khai mô hình trong thực tế.

Chương 3

Quá trình thực hiện

3.1 Chuẩn bị dữ liệu

3.1.1 Thu thập dữ liệu

Việc thu thập dữ liệu là bước đầu tiên và quan trọng nhất trong việc huấn luyện mô hình. Dữ liệu bao gồm các hình ảnh ô tô được chụp từ các nguồn khác nhau như camera giao thông, ảnh từ các nguồn công khai trên internet, hoặc tự chụp.

3.1.2 Xử lý và chuẩn bị dữ liệu

Dữ liệu sau khi thu thập được sẽ cần phải xử lý để đảm bảo tính đồng nhất và phù hợp cho mô hình. Các bước xử lý bao gồm:

- Cắt ảnh để loại bỏ các phần không cần thiết.
- Định kích thước lại hình ảnh để phù hợp với yêu cầu đầu vào của mô hình.
- Chuyển đổi ảnh sang định dạng phù hợp nếu cần thiết.

3.1.3 Tạo và gán annotation

Annotation là quá trình gán nhãn cho các đối tượng trong hình ảnh. Các nhãn này sẽ bao gồm thông tin về vị trí và loại đối tượng (trong trường hợp này là ô tô). Quá trình này có thể được thực hiện thủ công hoặc sử dụng các công cụ hỗ trợ như LabelImg, Labelbox.

3.1.4 Phân chia dữ liệu

Dữ liệu được phân chia thành ba tập: tập huấn luyện (training set), tập kiểm tra (validation set), và tập kiểm thử (test set). Thông thường, 70

3.2 Huấn luyện mô hình

Mô hình YOLOv8 sẽ được huấn luyện trên tập dữ liệu đã chuẩn bị. Quá trình huấn luyện bao gồm việc điều chỉnh các tham số của mô hình để tối ưu hóa độ chính xác trong việc nhận diện đối tượng. Các bước thực hiện bao gồm:

- Tải và tiền xử lý dữ liệu.
- Xây dựng mô hình YOLOv8 với các tham số đã cấu hình.
- Huấn luyện mô hình với tập dữ liệu huấn luyện.
- Theo dõi và điều chỉnh quá trình huấn luyện để tránh overfitting.

Chương 4

Quá Trình Huấn Luyện Mô Hình

4.1 Giới thiệu

Trong chương này, chúng tôi mô tả quy trình huấn luyện mô hình nhận diện biển số xe và ký tự trên biển số. Quy trình này bao gồm việc thu thập dữ liệu, chuẩn bị dữ liệu, và huấn luyện mô hình.

4.2 Thu Thập Dữ Liệu

4.2.1 Dữ Liệu Biển Số Xe

Dữ liệu để huấn luyện mô hình nhận diện biển số xe được thu thập từ Kaggle. Dataset này bao gồm các hình ảnh biển số xe với nhiều biến thể khác nhau, giúp mô hình học được các đặc điểm khác nhau của biển số. Bạn có thể truy cập dữ liệu này tại liên kết sau:

- Dữ liệu biển số xe: <https://www.kaggle.com/datasets/duydieunguyen/licenseplates>

Dữ liệu trên gồm khoảng hơn 3000 ảnh biển số xe. Mỗi ảnh có thể có một hoặc nhiều biển số được đánh labels sẵn.

4.2.2 Dữ Liệu Ký Tự

Để huấn luyện mô hình nhận diện ký tự trên biển số, chúng tôi sử dụng dataset ký tự được lưu trữ trên Google Drive. Dataset này bao gồm các ký



Hình 4.1: Ví dụ về hình ảnh biển số xe từ dataset.

tự và chữ số với nhiều kiểu dáng và kích thước khác nhau. Liên kết đến dataset này là:

- Dữ liệu ký tự: <https://drive.google.com/file/d/1bPux9J0e1mz-Jssx4XX1-wPGamaS8mI/view?usp=sharing>

0020	0030	0031	0032	0033	0034	0035	0036	0037	0038	0039	003A	0041	0042	0043
	0	1	2	3	4	5	6	7	8	9	:	A	B	C
0044	0045	0046	0047	0048	0049	004A	004B	004C	004D	004E	004F	0050	0051	0052
	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
0053	0054	0055	0056	0057	0058	0059	005A	005C	005F					
	S	T	U	V	W	X	Y	Z	\	-				

 **Taimienphi**
.vn

Hình 4.2: Ví dụ về hình ảnh các ký tự từ dataset.

Dữ liệu từ tập trên cũng hơn 3000 ảnh các ký tự khác nhau.

4.3 Chuẩn Bị Dữ Liệu

Dữ liệu từ cả hai nguồn đều được xử lý để phù hợp với yêu cầu của mô hình huấn luyện. Các bước chuẩn bị dữ liệu bao gồm:

- **Tiền xử lý dữ liệu:** Chúng tôi đã thực hiện các bước tiền xử lý sau:
 - Chuyển đổi hình ảnh về định dạng đồng nhất, ví dụ như thay đổi kích thước tất cả các hình ảnh về kích thước cố định hoặc tỉ lệ nhất quán.
 - Chia dữ liệu thành các tập huấn luyện, kiểm tra và validation để đảm bảo mô hình có thể được đánh giá chính xác.
- **Gán nhãn dữ liệu:** Đối với dữ liệu biển số xe:
 - Chúng tôi đã sử dụng định dạng nhãn phù hợp với mô hình YOLO, bao gồm các tọa độ của các bounding box và nhãn tương ứng.

Đối với dữ liệu ký tự:

- Chúng tôi đã gán nhãn các ký tự trên hình ảnh biển số bằng cách xác định vị trí và loại ký tự trong từng hình ảnh.
- **Mở rộng dữ liệu:** Để cải thiện khả năng tổng quát của mô hình và làm cho nó mạnh mẽ hơn với các biến thể của dữ liệu thực tế, chúng tôi đã áp dụng các kỹ thuật mở rộng dữ liệu sau:
 - **Xoay hình ảnh:** Chúng tôi đã xoay hình ảnh với các góc khác nhau (ví dụ: từ -15 đến 15 độ) để giúp mô hình nhận diện biển số xe và ký tự ở các góc khác nhau.
 - **Thêm nhiễu salt-and-pepper:** Để làm cho mô hình có thể xử lý các hình ảnh bị nhiễu, chúng tôi đã thêm nhiễu salt-and-pepper vào các hình ảnh. Kỹ thuật này giúp mô hình trở nên bền hơn với các hình ảnh bị nhiễu.
 - **Điều chỉnh độ tương phản:** Chúng tôi đã thay đổi độ tương phản của hình ảnh để mô hình có thể nhận diện các ký tự và biển số trong các điều kiện ánh sáng khác nhau.
 - **Điều chỉnh độ sáng:** Tương tự như điều chỉnh độ tương phản, việc thay đổi độ sáng giúp mô hình học được cách nhận diện trong các điều kiện ánh sáng thay đổi.

kết quả cuối cùng là gồm 2 tập dữ liệu với tổng số lượng dữ liệu mỗi tập là khoảng 10000 ảnh!



4.4 Huấn Luyện Mô Hình

Chúng tôi đã sử dụng các mô hình YOLOv8 khác nhau cho các nhiệm vụ cụ thể trong dự án nhận diện biển số xe và ký tự. Quá trình huấn luyện bao gồm các bước sau:

- **Chọn cấu hình mô hình:**

- Đối với việc nhận diện biển số xe, chúng tôi đã sử dụng mô hình YOLOv8 với khả năng phát hiện bounding box theo hướng (YOLOv8-OBB). Mô hình này phù hợp với các bài toán phát hiện đối tượng có góc nghiêng hoặc bị xoay.
- Đối với việc nhận diện ký tự trên biển số, chúng tôi đã sử dụng mô hình YOLOv8n (phiên bản nhẹ của YOLOv8) để phát hiện các ký tự.

- **Huấn luyện mô hình:**

- Chạy quá trình huấn luyện trên dữ liệu đã chuẩn bị cho từng mô hình. Việc này bao gồm theo dõi và điều chỉnh các tham số huấn luyện như tốc độ học và số epoch. Ở đây tôi lựa chọn 50epoch cho mô hình detect biển số và 40 epoch cho mô hình detect ký tự.
- Đối với mô hình YOLOv8-OBB, chúng tôi đã huấn luyện để phát hiện biển số xe với các bounding box có thể xoay.
- Đối với mô hình YOLOv8n, chúng tôi đã huấn luyện để phát hiện các ký tự trên biển số.

- **Đánh giá mô hình:**

- Sử dụng tập kiểm tra để đánh giá hiệu suất của từng mô hình. Đánh giá bao gồm việc kiểm tra độ chính xác của việc phát hiện các biển số xe và ký tự.
- Dựa trên kết quả đánh giá, điều chỉnh các tham số nếu cần thiết để cải thiện hiệu suất.

Hình 4.3 và Hình 4.4 minh họa quá trình huấn luyện các mô hình YOLOv8-OBB và YOLOv8n.

```
from ultralytics import YOLO

# tạo mô hình YOLOv8 OBB
model = YOLO("yolov8n-obb.yaml") # build a new model from yaml

model = model.to('cuda') # đưa mô hình lên GPU

# huấn luyện mô hình
model.train(data='dataset.yaml', epochs=50, imgsz=640, batch=16, amp=False)

metrics = model.val() # đánh giá mô hình
```

Hình 4.3: Ví dụ về quá trình huấn luyện mô hình YOLOv8-OBB cho việc phát hiện biển số xe.

```
from ultralytics import YOLO

# Tạo mô hình YOLOv8n OBB
model = YOLO("yolov8n.yaml") # build a new model from YAML

model = model.to("cuda") # đưa mô hình lên GPU

# huấn luyện mô hình
model.train(data="characterDetect.yaml", epochs=40, imgsz=640, batch=16, amp=False)

metrics = model.val() # đánh giá mô hình
```

Hình 4.4: Ví dụ về quá trình huấn luyện mô hình YOLOv8n cho việc phát hiện ký tự trên biển số.

Chương 5

Phân Tích Kết Quả Huấn Luyện Mô Hình

Trong chương này, chúng tôi phân tích kết quả của quá trình huấn luyện mô hình YOLOv8 được sử dụng để phát hiện biển số xe. Các chỉ số chính được xem xét bao gồm các giá trị mất mát (loss), độ chính xác (precision), độ nhạy (recall), và mAP (mean Average Precision).

5.1 Các Chỉ Số Đánh Giá

- **Mất Mát (Loss):** Các chỉ số mất mát bao gồm:
 - `train/box_loss`: Mất mát liên quan đến việc dự đoán bounding box.
 - `train/cls_loss`: Mất mát liên quan đến việc phân loại các đối tượng.
 - `train/dfl_loss`: Mất mát liên quan đến việc dự đoán phân phối xác suất.
- **Độ Chính Xác (Precision):** Đo lường tỷ lệ các phát hiện đúng trên tổng số các phát hiện được mô hình đưa ra.
- **Độ Nhạy (Recall):** Đo lường tỷ lệ các đối tượng thực sự được phát hiện so với tổng số đối tượng thực tế.
- **mAP (mean Average Precision):** Đo lường hiệu suất tổng thể của mô hình ở nhiều mức độ IoU (Intersection over Union).
 - `metrics/mAP50(B)`: mAP tại ngưỡng IoU 0.5.

- `metrics/mAP50-95(B)`: mAP trung bình tại các ngưỡng IoU từ 0.5 đến 0.95.

5.2 Đánh Giá Kết Quả Huấn Luyện

5.2.1 Phân Tích Mất Mát

Kết quả huấn luyện cho thấy sự giảm dần của các giá trị mất mát qua các epoch, cho thấy mô hình đang học và cải thiện khả năng phát hiện:

- `train/box_loss` giảm từ 2.0357 xuống 0.3349.
- `train/cls_loss` giảm từ 2.7472 xuống 1.1932.
- `train/dfl_loss` giảm từ 3.447 xuống 1.1932.

Sự giảm dần này cho thấy mô hình ngày càng chính xác hơn trong việc dự đoán bounding box, phân loại đối tượng và phân phối xác suất.

5.2.2 Hiệu Suất Mô Hình

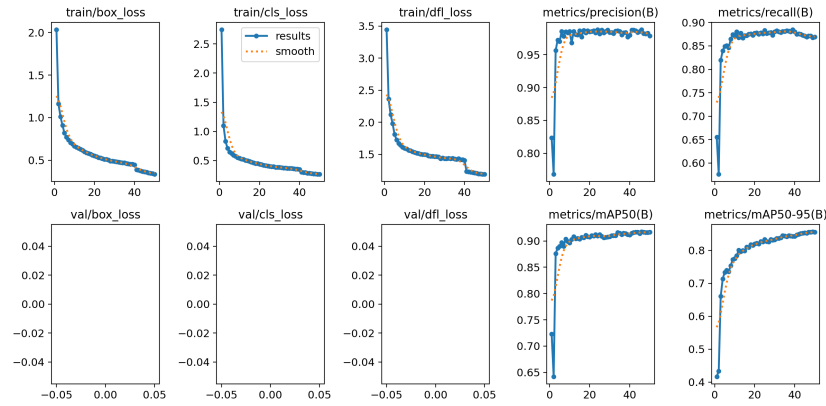
Các chỉ số độ chính xác và độ nhạy cho thấy sự cải thiện qua các epoch:

- `metrics/precision(B)` và `metrics/recall(B)` đều có xu hướng tăng, điều này cho thấy mô hình ngày càng phát hiện đúng và tìm thấy tất cả các đối tượng tốt hơn.
- `metrics/mAP50(B)` tăng từ 0.7229 lên 0.9785.
- `metrics/mAP50-95(B)` cũng tăng lên, cho thấy khả năng phát hiện của mô hình đang cải thiện ở nhiều mức độ IoU.

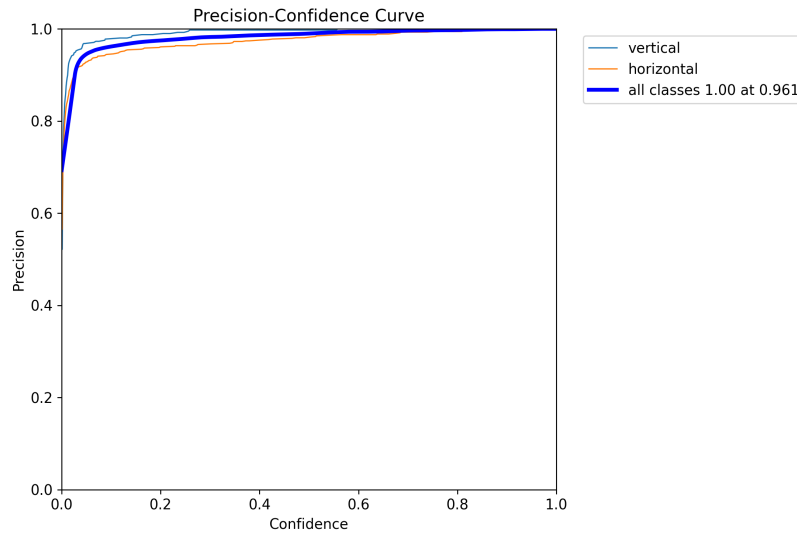
Những kết quả này chứng tỏ mô hình YOLOv8 đã đạt được hiệu suất tốt trong việc phát hiện biển số xe và có khả năng tổng quát tốt hơn với dữ liệu mới.

5.3 Hình Ảnh Minh Họa

Hình [5.1](#) và [5.2](#) minh họa quá trình huấn luyện và các chỉ số hiệu suất của mô hình:



Hình 5.1: Đường cong mất mát và các chỉ số đánh giá qua các epoch.



Hình 5.2: Đồ thị độ chính xác và độ nhạy của mô hình qua các epoch.

5.4 Kết Quả Đánh Giá Mô Hình

5.4.1 Kết Quả Phát Hiện Biển Số

Mô hình YOLOv8 đã cho kết quả tốt trong việc phát hiện biển số xe và nội dung biển số. Các hình ảnh dưới đây minh họa khả năng của mô hình trong việc phát hiện biển số và nhận diện nội dung của chúng.

Hình 5.3 và Hình 5.4 cho thấy mô hình đã phát hiện chính xác vị trí của biển số và đọc đúng nội dung của biển số xe. Điều này chứng tỏ mô hình có khả năng hoạt động hiệu quả trong điều kiện tốt.

5.4.2 Điểm Yếu và Các Tình Huống Khó

Các hình ảnh dưới đây minh họa những tình huống khó khăn mà mô hình gặp phải:

Mặc dù mô hình đã đạt được kết quả tốt trong nhiều trường hợp, nhưng vẫn còn một số điểm yếu cần được cải thiện, đặc biệt là trong các điều kiện ảnh không lý tưởng. Các vấn đề chính bao gồm:

- **Ảnh Bị Nghiêng Quá Nhiều:** Mô hình có thể gặp khó khăn trong việc phát hiện và nhận diện biển số khi biển số bị nghiêng quá nhiều.
- **Ảnh Mờ:** Các biển số bị mờ hoặc không rõ nét có thể làm giảm độ chính xác của mô hình.
- **Ảnh Bị Che Khuất:** Khi biển số bị che khuất bởi các đối tượng khác, mô hình có thể không phát hiện được hoặc nhận diện chính xác.

Những vấn đề này chỉ ra rằng mô hình cần được cải thiện để xử lý tốt hơn trong các điều kiện không lý tưởng. Các phương pháp mở rộng và cải thiện mô hình để xử lý các tình huống khó khăn này sẽ được xem xét trong các nghiên cứu tiếp theo.

5.5 Kết Luận và Hướng Đi Tiếp Theo

5.5.1 Kết Luận

Trong nghiên cứu này, chúng tôi đã phát triển và huấn luyện hai mô hình YOLOv8 cho các nhiệm vụ nhận diện biển số xe và ký tự trên biển số. Mô hình YOLOv8-OBBS đã được sử dụng hiệu quả để phát hiện biển số xe, trong khi mô hình YOLOv8-n được áp dụng để nhận diện ký tự trên biển số.

Kết quả huấn luyện cho thấy mô hình hoạt động tốt trong các điều kiện lý tưởng, với khả năng phát hiện và nhận diện biển số cũng như ký tự đạt hiệu quả cao. Tuy nhiên, mô hình còn gặp khó khăn trong các điều kiện không lý tưởng như biển số bị nghiêng, mờ, hoặc che khuất. Điều này cho thấy cần có những cải tiến để nâng cao độ chính xác và khả năng của mô hình trong điều kiện thực tế.

5.5.2 Hướng Đi Tiếp Theo

Để cải thiện hiệu suất của mô hình và tăng cường khả năng hoạt động trong các điều kiện thực tế, chúng tôi đề xuất một số phương án và hướng đi tiếp theo như sau:

- **Tăng Cường Dữ Liệu (Data Augmentation):**

- **Nghiêng Ảnh (Perspective Transform):** Áp dụng các biến đổi hình học để tạo ra các phiên bản của ảnh với các góc nghiêng khác nhau. Việc này giúp mô hình học cách nhận diện biển số từ nhiều góc nhìn khác nhau, đặc biệt khi biển số bị nghiêng hoặc nhìn từ các góc độ không thẳng hàng. Quá trình thực hiện bao gồm việc xác định các điểm góc trên biển số trong ảnh gốc và các điểm tương ứng trên ảnh đã nghiêng, sau đó tạo ma trận biến đổi và áp dụng nó lên ảnh.
- **Xoay Ảnh:** Tạo ra các phiên bản xoay của ảnh với các góc khác nhau (ví dụ: 15° , 30° , 45°) để mô hình có thể nhận diện biển số trong các góc khác nhau. Việc này giúp mô hình học cách nhận diện biển số từ các góc nhìn khác nhau, đặc biệt là khi biển số bị nghiêng.
- **Thêm Nhiều Salt & Pepper:** Thêm nhiều salt pepper vào ảnh để giúp mô hình học cách xử lý các ảnh bị nhiễu, cải thiện khả năng nhận diện trong điều kiện kém chất lượng. Bạn có thể điều chỉnh tỷ lệ nhiễu để mô phỏng các mức độ nhiễu khác nhau.
- **Điều chỉnh Tương Phản và Độ Sáng:** Áp dụng các thay đổi về tương phản và độ sáng (ví dụ: làm tối ảnh, làm sáng ảnh, tăng giảm độ tương phản) để mô hình trở nên linh hoạt hơn trong các điều kiện ánh sáng khác nhau. Điều này giúp mô hình có thể hoạt động tốt hơn trong các điều kiện ánh sáng yếu hoặc chói.
- **Crop và Zoom:** Cắt và phóng to các phần của ảnh (ví dụ: cắt một phần của biển số, zoom vào một phần cụ thể) để mô hình có thể học cách nhận diện biển số trong các điều kiện khác nhau. Kỹ thuật này giúp mô hình nhận diện các chi tiết nhỏ hơn của biển số.
- **Thay Đổi Kích Thước Ảnh:** Thay đổi kích thước của ảnh để mô hình có thể nhận diện biển số với các kích thước khác nhau. Ví dụ: tạo các phiên bản với kích thước ảnh nhỏ hơn hoặc lớn hơn để mô hình có thể làm việc với các biển số có kích thước khác nhau.
- **Phản Chiếu (Mirroring):** Tạo các phiên bản ảnh phản chiếu theo chiều ngang hoặc dọc để mô hình có thể nhận diện biển số trong các tình huống mà biển số có thể được nhìn từ góc độ khác.
- **Thay Đổi Màu Sắc (Color Jitter):** Áp dụng các thay đổi nhỏ

về màu sắc, bao gồm điều chỉnh độ bão hòa và hue của ảnh để mô hình có thể hoạt động trong các điều kiện màu sắc khác nhau.

- **Thêm Hiệu Ứng Mờ (Blurring Effects):** Thêm các hiệu ứng mờ (blur) như Gaussian blur hoặc motion blur vào ảnh để mô hình học cách nhận diện biển số trong các tình huống mà ảnh có thể bị mờ hoặc chuyển động.
- **Thay Đổi Tỷ Lệ (Scaling):** Thay đổi tỷ lệ của ảnh để mô hình có thể nhận diện biển số ở các tỷ lệ khác nhau. Điều này giúp mô hình học cách nhận diện biển số khi nó nằm ở các kích thước khác nhau trong ảnh.
- **Thay Đổi Độ Nghiêng (Perspective Transform):** Áp dụng các biến đổi hình học để mô hình học cách nhận diện biển số từ các góc nhìn khác nhau, đặc biệt là khi biển số bị nghiêng hoặc nhìn từ các góc khác nhau.

- **Cải Thiện Kiến Trúc Mô Hình:**

- **Sử Dụng Mô Hình Tiên Tiến Hơn:** Khám phá các phiên bản cải tiến của YOLO hoặc các mô hình khác như EfficientDet hay DetectoRS để cải thiện độ chính xác và khả năng phát hiện.
- **Tinh Chỉnh Hyperparameters:** Điều chỉnh các tham số như tốc độ học, kích thước batch, và các tham số khác để tối ưu hóa hiệu suất của mô hình.

- **Thu Thập Dữ Liệu Mới:** Mở rộng tập dữ liệu với các ví dụ từ các tình huống khó khăn như biển số bị nghiêng, mờ, hoặc che khuất. Điều này giúp mô hình học được các đặc điểm của các tình huống này và cải thiện khả năng nhận diện.

- **Tăng Cường Xử Lý Hậu Xử Lý:**

- **Sử Dụng Các Kỹ Thuật Hậu Xử Lý:** Áp dụng các kỹ thuật hậu xử lý như lọc và làm sạch để cải thiện chất lượng dự đoán từ mô hình.

Bằng cách thực hiện các cải tiến trên, chúng tôi hy vọng sẽ nâng cao độ chính xác và khả năng hoạt động của mô hình trong các điều kiện thực tế, góp phần vào sự phát triển của các hệ thống nhận diện biển số xe và ký tự chính xác và hiệu quả hơn.