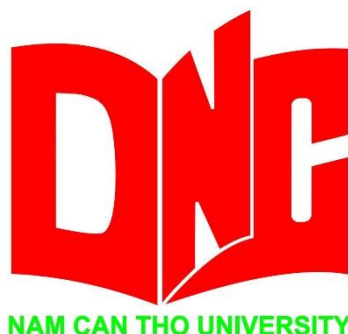


TRƯỜNG ĐẠI HỌC NAM CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN



HỌ VÀ TÊN TÁC GIẢ
LÊ VĂN HỮU

TÊN ĐỀ TÀI
HUẤN LUYỆN MÔ HÌNH THỊ GIÁC MÁY TÍNH
NHẬN DẠNG BIỂN SỐ XE

KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC

Ngành: Công Nghệ Thông Tin

Mã số ngành: 7480201

Tháng 6 năm 2024

TRƯỜNG ĐẠI HỌC NAM CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN



HỌ VÀ TÊN TÁC GIẢ

LÊ VĂN HỮU

MSSV: 201534

DH20TIN03

TÊN ĐỀ TÀI:

**HUẤN LUYỆN MÔ HÌNH THỊ GIÁC MÁY TÍNH
NHẬN DẠNG BIỂN SỐ XE**

KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC

NGÀNH: CÔNG NGHỆ THÔNG TIN

Mã số ngành: 7480201

CÁN BỘ HƯỚNG DẪN

TS. NGÔ VIỆT THỊNH

Tháng 6 năm 2024

CHẤP THUẬN CỦA HỘI ĐỒNG

Khóa luận “”, do sinh viênthực hiện dưới sự hướng dẫn của..... Khóa luận đã báo cáo và được Hội đồng chấm khóa luận thông qua ngày

Ủy viên

Thư ký

(Ký tên)

(Ký tên)

ThS.GVC. Lê Đức Thắng

TS. Ngô Viết Thịnh

Phản biện 1

Phản biện 2

ThS.GVC. Đoàn Hòa Minh

ThS. Trương Hùng Chen

Cán bộ hướng dẫn

Chủ tịch Hội đồng

TS. Ngô Viết Thịnh

ThS. Võ Văn Phúc

LỜI CẢM TẠ

Em xin chân thành cảm ơn các thầy cô khoa Công nghệ thông tin đã trang bị cho em những kiến thức quý báu trong suốt quá trình học tập và làm đề tài tại Trường Đại học Nam Cần Thơ. Nhờ công lao giảng dạy, chỉ bảo tận tình của các thầy cô mà em đã có được những kiến thức chuyên ngành về công nghệ thông tin để thực hiện đề tài này. Em xin gửi lời chúc sức khỏe và lời chào trân trọng nhất đến các thầy cô của khoa Công nghệ thông tin, Trường Đại học Nam Cần Thơ. Với sự dạy dỗ tận tình qua các môn học từ cơ bản đến nâng cao của các thầy cô, em nay đã có đủ kiến thức và cơ sở để hoàn thành đề tài này.

Đặc biệt, em xin gửi lời cảm ơn chân thành nhất đến giảng viên TS. Ngô Viết Thịnh. Trong khoảng thời gian thực hiện đề tài, em cảm ơn thầy đã quan tâm, hướng dẫn tận tình, chi tiết và đầy đủ, giúp em có đủ kiến thức và khả năng áp dụng để hoàn thành tốt đề tài lần này.

Tiếp theo em xin chân thành cảm ơn Ban Giám hiệu nhà trường đã tạo điều kiện thuận lợi và môi trường học tập tốt nhất cho em trong suốt quá trình học tập và nghiên cứu. Sự hỗ trợ và tạo điều kiện của nhà trường là nền tảng vững chắc giúp em hoàn thành đề tài này.

Em xin gửi lời cảm ơn sâu sắc đến đơn vị nhà xe, nơi đã cung cấp dữ liệu quý báu để em có thể tiến hành nghiên cứu và thực hiện đề tài. Sự hỗ trợ của đơn vị nhà xe đã đóng góp vào thành công của đề tài này.

Mặc dù đã cố gắng nỗ lực và quyết tâm thực hiện đề tài, tuy nhiên bài báo cáo chắc chắn không tránh khỏi thiếu sót, em rất mong nhận được những ý kiến đóng góp quý báu của thầy cô và các bạn để kiến thức của em được hoàn thiện hơn.

Em xin chân thành cảm ơn, và kính chúc quý thầy cô dồi dào sức khỏe.

Cần Thơ, Ngày....Tháng....Năm 2024

Sinh viên thực hiện

Lê Văn Hữu

LỜI CAM KẾT

Em xin cam kết những dữ liệu và kết quả nghiên cứu là trung thực, không sao chép bất kỳ nguồn nào khác. Ngoài ra, trong báo cáo có sử dụng một số nguồn tài liệu tham khảo đã được trích dẫn nguồn và chú thích rõ ràng. Em xin hoàn toàn chịu trách nhiệm trước bộ môn, khoa và nhà trường về sự cam đoan này.

Em xin cam kết khóa luận này được hoàn thành dựa trên các kết quả nghiên cứu của em trong khuôn khổ của đề tài/dự án "*Huấn Luyện Mô Hình Thị Giác Máy Tính Nhận Dạng Biển Số Xe*". Dự án có quyền sử dụng kết quả của khóa luận này để phục vụ cho dự án.

Cần Thơ, Ngày....Tháng....Năm 2024

Sinh viên thực hiện

Lê Văn Hữu

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the entire width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

Giảng viên hướng dẫn

111

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the entire width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

Giảng viên phản biện

iv

[illegible]

Giảng viên phản biện

V

MỤC LỤC

CHƯƠNG 1 GIỚI THIỆU.....	4
1.1. Lý do chọn đề tài.....	4
1.2. Mục tiêu nghiên cứu.....	5
1.3. Phạm vi nghiên cứu.....	5
CHƯƠNG 2 CƠ SỞ LÝ LUẬN VÀ PHƯƠNG PHÁP NGHIÊN CỨU	6
2.1. Cơ sở lý luận.....	6
2.1.1. Giới thiệu về YOLO	6
2.1.2. Các phiên bản của YOLO.....	7
2.1.3. Ưu điểm của YOLOv8.....	8
2.1.4. Giới thiệu về OCR (Optical Character Recognition)	10
2.1.5. Kết hợp YOLO và EasyOCR trong nhận dạng biển số xe.....	11
2.2. Phương pháp nghiên cứu.....	12
2.2.1 Thu thập và chuẩn bị dữ liệu	13
2.2.2 Kiến trúc mạng YOLOv8.....	13
2.2.3. Kiến trúc EasyOCR.....	14
2.2.4. Huấn luyện mô hình	14
CHƯƠNG 3 THỰC NGHIỆM.....	17
3.1. Môi trường thực nghiệm.....	17
3.2. Chuẩn bị dữ liệu.....	18
3.3. Phân bổ dữ liệu.....	18
3.4. Huấn luyện mô hình	22
3.5 Thử nghiệm và đánh giá.....	27
3.5.1. Thử nghiệm chương trình.....	28
3.5.1. Đánh giá chương trình	28
CHƯƠNG 4 KẾT QUẢ NGHIÊN CỨU.....	29
4.1. Kết quả huấn luyện và kiểm tra mô hình.....	29
4.2. Phân tích kết quả.....	29
CHƯƠNG 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	31
5.1. Kết luận	31
5.2. Hướng phát triển của đề tài	32
TÀI LIỆU THAM KHẢO.....	33

DANH MỤC BẢNG

Bảng 2.1: Các phiên bản của YOLO.....	8
Bảng 3.1: Phần cứng dùng để huấn luyện mô hình.....	17
Bảng 3.2 : Yêu cầu môi trường.....	17
Bảng 3.3: Lưu trữ và quản lý các tập dữ liệu	19
Bảng 3.4: Phân bổ ảnh và label.....	19

DANH MỤC HÌNH

Hình 2.1: Ví dụ về nhận dạng YOLO	7
Hình 2.2: Cải tiến của YOLO so với các phiên bản trước đó.....	9
Hình 2.3: Phân đoạn ký tự	11
Hình 2.4: Ảnh gốc	12
Hình 2.5: Ảnh đã nhận diện	12
Hình 2.6: Mô hình mạng lưới neural đơn giản	16
Hình 3.1.: Nội dung file dataset.yaml	20
Hình 3.2: Bộ dữ liệu	21
Hình 3.4: Tiến hành train dữ liệu	23
Hình 3.5: Biểu đồ kết quả sau khi train	23
Hình 3.6: Lưu đồ chi tiết của hệ thống nhận dạng biển số xe.....	27
Hình 3.7: Thử nghiệm chương trình nhận dạng	28

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Giải thích
YOLO	You Only Look Once
FC	full-connected layers
Conv	convolutional layers
CNN	Convolutional Neural Network
IoU	Intersection over Union
val	Validation
DFL_loss	Distribution Focal Loss
Cls_loss	Classification Loss
Box_loss	Bounding boxes loss
OCR	Optical Character Recognition
SIFT	Scale-Invariant Feature Transform
SURF	Speeded Up Robust Features
HOG	Histogram of Oriented Gradients
CNNs	Convolutional Neural Networks
Faster R-CNN	Region-based Convolutional Neural Network

CHƯƠNG 1

GIỚI THIỆU

1.1. Lý do chọn đề tài

- Thị giác máy tính, hay Computer Vision, là một lĩnh vực của trí tuệ nhân tạo (AI) và khoa học máy tính, tập trung vào việc cho máy tính khả năng hiểu và diễn giải thông tin từ hình ảnh số hoặc video.

- Việc chọn đề tài "*Huấn luyện mô hình thị giác máy tính nhận dạng biển số xe*" xuất phát từ nhiều lý do phù hợp bao gồm:

- Tiềm năng thị trường: Với sự gia tăng nhanh chóng của số lượng phương tiện giao thông, nhu cầu về các giải pháp quản lý và giám sát giao thông ngày càng trở nên cấp thiết. Hệ thống nhận dạng biển số xe tự động có tiềm năng lớn trong việc cải thiện hiệu quả của các hệ thống giám sát giao thông, hệ thống bãi giữ xe, kiểm soát an ninh, và thực thi pháp luật. Điều này tạo ra một thị trường rộng lớn và đầy tiềm năng cho các giải pháp nhận dạng biển số xe.

- Nhu cầu thị trường: Chủ các bãi đỗ xe, các cơ quan quản lý giao thông, đều đối mặt với nhiều thách thức trong việc quản lý và giám sát phương tiện. Một hệ thống nhận dạng biển số xe hiệu quả sẽ giúp họ giải quyết các vấn đề như: kiểm soát ra vào, quản lý vi phạm giao thông, theo dõi và truy xuất thông tin phương tiện, quản lý bãi đỗ xe tự động, và nhiều tác vụ khác. Phần mềm nhận dạng biển số xe sẽ đáp ứng nhu cầu này, giúp tối ưu hóa quá trình quản lý và giám sát.

- Tính ứng dụng rộng rãi: Hệ thống nhận dạng biển số xe không chỉ áp dụng trong giao thông mà còn có thể được sử dụng trong nhiều lĩnh vực khác như: an ninh khu vực, quản lý ra vào tại các khu công nghiệp, khu dân cư, bãi đỗ xe công cộng và tư nhân. Sự đa dạng trong ứng dụng giúp mở rộng phạm vi sử dụng của công nghệ này.

- Sự tiện lợi: Hệ thống nhận dạng biển số xe tự động giúp giảm bớt khối lượng công việc cho nhân viên quản lý và giám sát, đồng thời tăng cường độ chính xác và hiệu quả trong việc nhận dạng và xử lý thông tin biển số xe. Điều này mang lại sự tiện lợi cho cả người quản lý và người sử dụng dịch vụ, đồng thời tăng cường tính an toàn và bảo mật cho các khu vực được giám sát.

- Tính hiệu quả: Công nghệ nhận dạng biển số xe sử dụng các thuật toán thị giác máy tính tiên tiến và trí tuệ nhân tạo để tự động hóa quá trình nhận dạng và xử lý thông tin biển số. Điều này giúp tiết kiệm thời gian, tài nguyên và công sức, cải thiện hiệu suất kinh doanh và giảm thiểu sai sót. Đồng thời, việc tự động hóa này cũng giúp tăng cường khả năng phát hiện và xử lý vi phạm giao thông một cách nhanh chóng và chính xác.

Chính vì vậy thách thức lớn đặt ra là phải xây dựng một hệ thống nhận diện có độ chính xác cao và khả năng hoạt động ổn định trong mọi điều kiện môi trường không lường trước được nên em đã quyết định chọn đề tài "*Huấn luyện mô hình thị giác máy tính nhận dạng biển số xe*" làm đề án của mình.

1.2. Mục tiêu nghiên cứu

- Mục tiêu của đề tài là nghiên cứu và đánh giá hiệu suất của một hệ thống nhận diện biển số xe máy dựa trên các công nghệ nhận diện bằng YOLO và nhận diện ký tự sử dụng EasyOCR. Đồng thời, đề tài cũng nhằm xây dựng một cơ sở dữ liệu phong phú và đa dạng để phục vụ cho việc huấn luyện và đánh giá hệ thống, cũng như hỗ trợ các nghiên cứu sau này. Bên cạnh đó, mục tiêu còn bao gồm phát triển một mô hình học sâu hiệu quả cho việc nhận diện biển số xe máy, và kiểm tra, đánh giá hiệu suất của hệ thống trong các điều kiện thực tế, đảm bảo rằng hệ thống hoạt động ổn định và chính xác trong môi trường sử dụng thực tế.

1.3. Phạm vi nghiên cứu

Vì vấn đề vị trí đang học tập và thời gian thực hiện luận văn cũng như kinh phí còn hạn chế nên đề tài "*Huấn luyện mô hình thị giác máy tính nhận dạng biển số xe*" chỉ có thể thực hiện thu thập bộ dữ liệu với phạm vi tại trường Đại Học Nam Cần Thơ. Về huấn luyện mô hình được thực hiện tại nhà trên máy tính cá nhân sử dụng các công cụ và mã nguồn mở hỗ trợ huấn luyện mô hình.

CHƯƠNG 2

CƠ SỞ LÝ LUẬN VÀ PHƯƠNG PHÁP NGHIÊN CỨU

2.1. Cơ sở lý luận

2.1.1. Giới thiệu về YOLO

YOLO (You Only Look Once) mô hình này lần đầu tiên được giới thiệu bởi Joseph Redmon và cộng sự vào năm 2016. Là mô hình mạng CNN (Convolutional Neural Network) cho phát hiện, nhận dạng, phân loại đối tượng phổ biến được biết đến với tốc độ nhanh và độ chính xác cao. Yolo được tạo ra từ việc kết hợp giữa các convolutional layers (Conv) và connected layers. Trong đó các convolutional layers sẽ trích xuất ra các đặc trưng (feature) của ảnh, còn full-connected layers (FC) sẽ dự đoán ra xác suất đó và tọa độ của đối tượng.

Đầu vào của mô hình là một ảnh, mô hình sẽ nhận dạng ảnh đó có đối tượng nào hay không, sau đó sẽ xác định tọa độ của đối tượng trong bức ảnh. Ảnh đầu vào được chia thành $S \times S$ ô thường sẽ là $3 \times 3, 7 \times 7, 9 \times 9, \dots$

Với input là 1 ảnh, đầu ra mô hình là một ma trận có kích thước: $S \times S \times (5 \times N + M)$ với số lượng tham số mỗi ô là $(5 \times N + M)$

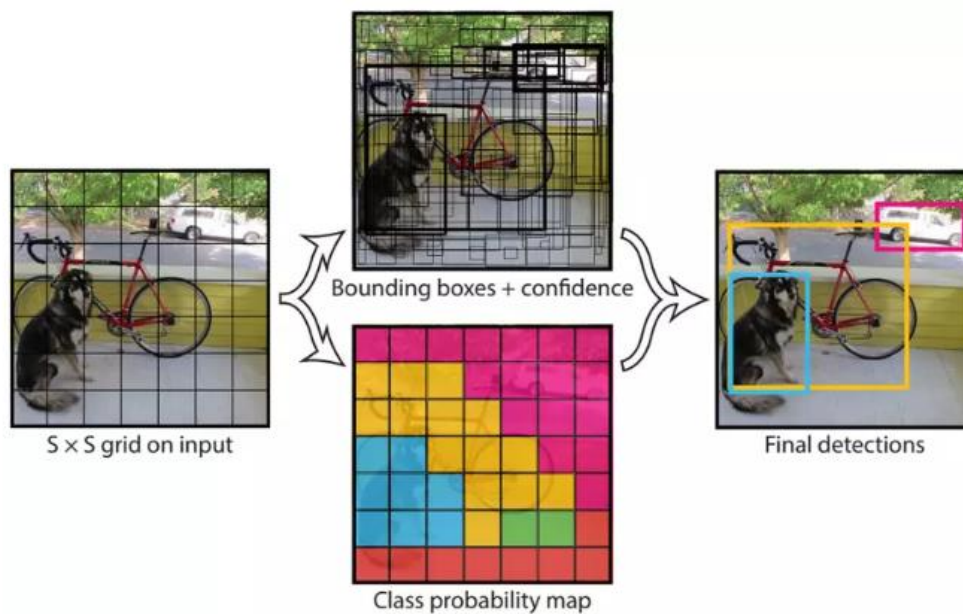
Với N và M lần lượt là số lượng Box và Class mà mỗi ô cần dự đoán.

Với 5 là dự đoán mỗi bounding box: 4 tọa độ ($x, y, width, height$) đại diện cho vị trí và kích thước của bounding box và 1 prediction (độ tin cậy). Prediction được định nghĩa $Pr(\text{Object}) \times IOU(\text{pred}, \text{truth})$

IoU là thước đo sự chồng chéo giữa hộp giới hạn dự đoán và hộp giới hạn thực tế. IoU càng cao, mô hình càng chính xác.

Công thức tính IoU:

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$



Hình 2.1: Ví dụ về nhận dạng YOLO
(Nguồn: Viblo. Tìm hiểu về YOLO trong bài toán real-time object detection)

Ví dụ:

Hình ảnh trên được chia thành:

Số lượng ô ($S \times S$): 7×7

Mỗi ô cần dự đoán 2 bounding box mỗi ô (N): 2

Cần dự đoán 3 object classes: con chó, ô tô và xe đạp (M) : 3

Số lượng tham số mỗi ô ($5 \times N + M$): $\Rightarrow (5 \times 2 + 3)$ mỗi ô có 13 tham số

Kích thước ma trận đầu ra (output) là $7 \times 7 \times (5 \times 2 + 3) = 637$ tham số

Số lượng bounding box: $7 \times 7 \times 2 = 98$

2.1.2. Các phiên bản của YOLO

Phiên bản	Năm ra đời	Phương pháp/ Cải tiến	Đặc điểm
YOLOv1	2015 Joseph Redmon	Chia hình ảnh thành lưới $S \times S$	Nhanh, nhận diện thời gian thực, hạn chế với các đối tượng nhỏ và gần nhau.
YOLOv2 (YOLO 9000)	2016	Batch Normalization, sử dụng Anchor Boxes, Multi-Scale Training	Tăng độ chính xác, YOLO9000 dự đoán hơn 9000 lớp đối tượng.

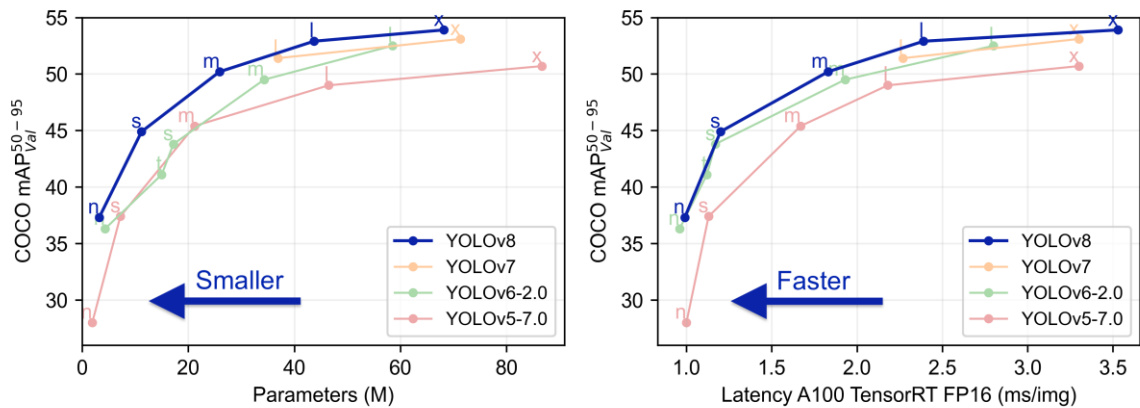
Phiên bản	Năm ra đời	Phương pháp/ Cải tiến	Đặc điểm
YOLOv3	2018	Kiến trúc Darknet-53, dự đoán ở ba mức phân giải khác nhau.	Cải thiện nhận diện đối tượng nhỏ, sử dụng Binary Cross-Entropy Loss.
YOLOv4	2020 bởi Alexey Bochkovskiy	CSPDarknet53, Bag of Freebies (BoF), Bag of Specials (BoS).	Tăng hiệu suất, giảm chi phí tính toán.
YOLOv5	2020 bởi Ultralytics	Xây dựng trên PyTorch, Data Augmentation (Mosaic, mixup).	Mô hình nhẹ, nhanh, dễ triển khai.
YOLOv6	2021	Tối ưu hóa kiến trúc, áp dụng Pruning và Quantization	Giảm kích thước mô hình, tăng tốc độ suy luận.
YOLOv7	2022	Tối ưu hóa kiến trúc mạng, kỹ thuật huấn luyện mới.	Cân bằng tốt giữa tốc độ và độ chính xác.
YOLOv8	2023	Sử dụng CSP Darknet cải tiến, nâng cao khả năng trích xuất đặc trưng. Cải thiện nhận diện đối tượng ở các kích thước khác nhau. Prediction Head: Các prediction head chuyên biệt cho phát hiện và phân loại đối tượng. Tối ưu hóa hiện đại giúp tăng tốc độ xử lý và giảm kích thước mô hình.	Tăng cường hiệu suất, độ chính xác và khả năng xử lý đối tượng đa dạng.

Bảng 2.1: Các phiên bản của YOLO

2.1.3. Ưu điểm của YOLOv8

YOLOv8 là một mô hình tiên tiến và hiện đại, kế thừa thành công của các phiên bản YOLO trước đó, đồng thời mang đến những tính năng và cải tiến mới nhằm tăng cường hiệu suất và tính linh hoạt. Với thiết kế để hoạt động nhanh, chính xác và dễ sử dụng, YOLOv8 là lựa chọn hoàn hảo cho nhiều nhiệm vụ như phát

hiện và theo dõi đối tượng, phân đoạn đối tượng, phân loại hình ảnh, ước tính tư thế và nhận dạng đối tượng, đặc biệt là trong bối cảnh nhận dạng biển số xe.



Hình 2.2: Cải tiến của YOLO so với các phiên bản trước đó (Nguồn: HackMD. Mô Hình Nhận Dạng Vật Thể YOLOv8.)

- Lý do lựa chọn YOLOv8

Hiệu suất cao: YOLOv8 mang lại hiệu suất cao trong việc phát hiện và nhận dạng đối tượng với tốc độ và độ chính xác vượt trội, phù hợp với nhu cầu thực tiễn của các ứng dụng thời gian thực. Điều này đặc biệt quan trọng trong nhận dạng biển số xe, nơi yêu cầu xử lý nhanh chóng và chính xác.

Dễ sử dụng: Với thiết kế giao diện thân thiện và dễ sử dụng, YOLOv8 giúp tiết kiệm thời gian và công sức trong quá trình triển khai và tích hợp. Điều này cho phép các nhà phát triển và kỹ sư dễ dàng áp dụng mô hình vào các ứng dụng thực tế mà không cần quá nhiều kinh nghiệm về lập trình.

Tính linh hoạt: YOLOv8 hỗ trợ nhiều nhiệm vụ thị giác máy tính khác nhau, từ phát hiện đối tượng đến phân đoạn đối tượng và ước tính tư thế, đáp ứng đa dạng yêu cầu của các ứng dụng thực tiễn. Đặc biệt, trong nhận dạng biển số xe, YOLOv8 có thể xử lý các tình huống phức tạp như biển số bị che khuất một phần hoặc biển số bị biến dạng.

Tiếp nối thành công: YOLOv8 kế thừa và phát triển dựa trên nền tảng của các phiên bản YOLO trước đó, vốn đã được chứng minh là hiệu quả và đáng tin cậy trong cộng đồng nghiên cứu và ứng dụng thị giác máy tính. Điều này đảm bảo rằng YOLOv8 có độ tin cậy cao và hiệu suất ổn định.

Cải tiến công nghệ: YOLOv8 tích hợp các cải tiến mới nhất trong lĩnh vực học sâu và thị giác máy tính, đảm bảo khả năng cạnh tranh với các framework khác như SSD, Faster R-CNN, và Mask R-CNN. Điều này giúp YOLOv8 duy trì vị thế hàng đầu trong các ứng dụng nhận dạng đối tượng, bao gồm nhận dạng biển số xe.

Cộng đồng hỗ trợ: Với một cộng đồng lớn mạnh và tài liệu phong phú, YOLOv8 giúp người dùng dễ dàng tìm kiếm hỗ trợ và cập nhật các kiến thức mới nhất. Điều này rất hữu ích cho việc giải quyết các vấn đề cụ thể và tối ưu hóa mô hình cho nhận dạng biển số xe.

Với những lý do trên, YOLOv8 là lựa chọn hợp lý và hiệu quả cho đề tài *"Huấn luyện mô hình thị giác máy tính nhận dạng biển số xe"*, đảm bảo mang lại kết quả tối ưu và đáp ứng tốt các yêu cầu nghiên cứu.

2.1.4. Giới thiệu về OCR (Optical Character Recognition)

Công nghệ OCR (Optical Character Recognition): Là một công nghệ nhận diện ký tự trên máy tính, công nghệ này giúp tách riêng phân chữ và số ra khỏi thông nền đồng thời chuyển nó về dạng ký tự để trả về. Trong đề tài, sau khi đã biết có sự xuất hiện hay không của biển và vị trí của nó, OCR đóng vai trò cắt ra được ký tự trong biển và chuyển nó về dạng ký tự trong máy tính để lưu trữ.

OCR hoạt động bằng cách phân tích hình ảnh và nhận dạng các mẫu ký tự. Các bước chính bao gồm tiền xử lý hình ảnh, phân đoạn ký tự, trích xuất đặc trưng, và phân loại ký tự. Các thuật toán OCR hiện đại thường sử dụng các mạng nơ-ron sâu để cải thiện độ chính xác của việc nhận dạng.



Hình 2.3: Phân đoạn ký tự

Có nhiều công nghệ và thư viện OCR phổ biến như Tesseract, EasyOCR, và các API OCR của các dịch vụ như Google Cloud Vision, Microsoft Azure Cognitive Services. Tesseract là một trong những công cụ OCR mã nguồn mở phổ biến nhất, được phát triển bởi Google

2.1.5. Kết hợp YOLO và EasyOCR trong nhận dạng biển số xe

Kết hợp giữa YOLO (You Only Look Once) và EasyOCR trong nhận dạng biển số xe có thể tạo ra một hệ thống mạnh mẽ và hiệu quả. Dưới đây là cách kết hợp hai công nghệ này để nhận dạng biển số xe:

- **Sử dụng YOLO để phát hiện biển số xe:**

Mô hình YOLO có thể được huấn luyện trên tập dữ liệu chứa ảnh các biển số xe cùng với nhãn tương ứng.

Sau khi huấn luyện, YOLO có khả năng phát hiện và định vị vị trí của biển số xe trên hình ảnh hoặc video.

- **Sử dụng EasyOCR để nhận dạng ký tự trên biển số xe:**

Sau khi YOLO đã xác định vị trí của biển số xe trên hình ảnh, vùng ảnh chứa biển số được cắt ra.

Các vùng ảnh này sau đó được đưa vào EasyOCR để nhận dạng và trích xuất ký tự.

EasyOCR sử dụng mạng nơ-ron sâu để nhận dạng ký tự với độ chính xác cao.

- **Kết hợp kết quả từ YOLO và EasyOCR:**

Kết quả từ YOLO (vị trí của biển số xe) và EasyOCR (chữ và số trên biển số) được kết hợp để tạo thành thông tin hoàn chỉnh về biển số xe.

Thông tin này có thể được sử dụng cho các mục đích như theo dõi xe, kiểm soát giao thông, bảo mật và giám sát an ninh.

Kết hợp YOLO và EasyOCR giúp cải thiện độ chính xác và hiệu suất trong quá trình nhận dạng biển số xe. Đồng thời, việc sử dụng cả hai công nghệ này cũng mang lại tính linh hoạt và đa dạng trong xử lý hình ảnh và nhận dạng ký tự.



Hình 2.4: Ảnh gốc



Hình 2.5: Ảnh đã nhận diện

2.2. Phương pháp nghiên cứu

- *Phương pháp nghiên cứu quan sát*: Tiếp xúc trực tiếp với đối tượng thực tế để thu thập số liệu, thông qua phương tiện quan sát trực tiếp như nghe, xem hoặc sử dụng các phương tiện ghi âm ghi hình, phương pháp này giúp ghi nhận sự việc đang xảy ra một cách trực tiếp, ít tốn kém và ít gây phản ứng từ đối tượng khảo sát.

- *Phương pháp nghiên cứu nghiên cứu tài liệu*: Bao gồm việc thu thập và phân tích các tài liệu khoa học, báo cáo nghiên cứu và các nguồn thông tin khác liên quan đến kỹ thuật huấn luyện mô hình nhận dạng biển số xe.

- *Một số phương pháp nhận diện ảnh phổ biến hiện có như*:

- + **Nhận diện dựa trên đặc trưng (Feature-based methods)**: Sử dụng các đặc trưng hoặc thuộc tính của đối tượng để nhận diện. Các phương pháp này thường dựa trên việc tạo ra và so khớp các điểm đặc trưng, như SIFT (Scale-Invariant Feature Transform), SURF (Speeded Up Robust Features), hoặc HOG (Histogram of Oriented Gradients).

- **Phương pháp dựa trên mạng nơ-ron (Neural network-based methods)**: Sử dụng mạng nơ-ron để tự động học và nhận dạng vật thể trong ảnh. Các kiến trúc

phổ biến trong phương pháp này bao gồm Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), và YOLO (You Only Look Once).

- **Nhận diện bằng phương pháp học sâu (Deep learning methods):** Sử dụng các mạng neural sâu để tự động học và nhận diện đối tượng trong ảnh hoặc video. Các phương pháp này bao gồm mạng neural convolutional (CNN) như YOLO (You Only Look Once), SSD (Single Shot MultiBox Detector), Faster R-CNN (Region-based Convolutional Neural Network) và Mask R-CNN.

- **Phương pháp kết hợp (Hybrid methods):** Kết hợp các phương pháp trên để cải thiện hiệu suất và độ chính xác của hệ thống nhận diện.

Để thực hiện nghiên cứu về đề tài “*Huấn luyện mô hình thị giác máy tính nhận dạng biển số xe*”: em sử dụng phương pháp nghiên cứu kết hợp

2.2.1 Thu thập và chuẩn bị dữ liệu

- Thu thập ảnh: Tập hợp các ảnh chứa biển số xe từ các nguồn khác nhau.
- Tiền xử lý ảnh: Các ảnh được tiền xử lý để chuẩn hóa kích thước, màu sắc, và cắt các vùng quan trọng như biển số xe.
- Gắn nhãn ảnh: Mỗi ảnh được gắn nhãn với các bounding box xác định vị trí và lớp của biển số xe.

2.2.2 Kiến trúc mạng YOLOv8

Mô tả chi tiết về cấu trúc mạng YOLOv8 bao gồm các thành phần chính sau:

- **Backbone:** Đây là phần của mạng sử dụng để trích xuất đặc trưng từ ảnh đầu vào. Các kiến trúc backbone thường là các mạng nơ-ron sâu như Darknet, ResNet, hoặc EfficientNet, được điều chỉnh để phù hợp với yêu cầu của bài toán nhận dạng biển số xe.

- **Neck:** Phần này chịu trách nhiệm kết nối các đặc trưng từ backbone để chuẩn bị cho giai đoạn dự đoán. Các phương pháp thường được áp dụng ở đây là các phép gộp hoặc các module attention nhằm cải thiện khả năng biểu diễn.

- **Head:** Đây là phần cuối cùng của mạng, có nhiệm vụ dự đoán các bounding box và các lớp của các đối tượng trong ảnh. Head thường bao gồm các lớp convolutional để dự đoán các thông tin liên quan đến đối tượng, bao gồm cả vị trí và loại.

2.2.3. Kiến trúc EasyOCR

EasyOCR sử dụng mạng nơ-ron tích chập (thAI) và mạng nơ-ron hồi quy (RNN) để xử lý và nhận dạng ký tự từ hình ảnh. Mô hình này có thể được huấn luyện và tùy chỉnh dễ dàng để đáp ứng các yêu cầu cụ thể của người dùng

2.2.4. Huấn luyện mô hình

Quá trình huấn luyện mô hình YOLOv8 là một phần quan trọng trong việc đảm bảo mô hình có thể nhận dạng biển số xe một cách chính xác và hiệu quả. Phần này bao gồm các bước cấu hình, thực hiện huấn luyện và tối ưu hóa mô hình để đạt được kết quả tốt nhất.

Cấu hình các tham số huấn luyện

Trước khi bắt đầu huấn luyện, việc cấu hình các tham số huấn luyện là cực kỳ quan trọng để đảm bảo mô hình đạt được hiệu suất cao. Các tham số chính cần được cân nhắc và điều chỉnh bao gồm:

- **Learning rate (tốc độ học):** Đây là một trong những tham số quan trọng nhất, quyết định tốc độ cập nhật trọng số của mô hình trong quá trình huấn luyện. Thường được khởi đầu ở một giá trị nhỏ và có thể giảm dần theo thời gian để đảm bảo hội tụ tối ưu.

- **Batch size (kích thước batch):** Là số lượng ảnh được sử dụng trong mỗi lần cập nhật gradient. Lựa chọn batch size phù hợp giữa hiệu suất tính toán và khả năng chứa đựng của bộ nhớ.

- **Epochs (số lượng epoch):** Là số lượng lần mô hình sẽ huấn luyện trên toàn bộ tập dữ liệu. Cần thiết phải đủ lớn để mô hình học được các đặc trưng tổng quát từ dữ liệu nhưng cũng không quá lớn để tránh quá khớp (overfitting).

- **Optimizer (bộ tối ưu hóa):** Sử dụng các thuật toán tối ưu hóa như Adam, SGD để điều chỉnh trọng số mô hình dựa trên gradient của hàm mất mát.

- **Loss function (hàm mất mát):** Được sử dụng để đánh giá sự khác biệt giữa dự đoán của mô hình và các giá trị thực tế. Hàm mất mát đa mục tiêu trong YOLOv8 bao gồm hai thành phần chính:

+ **Localization loss** (hàm mất mát về vị trí): Đo lường sai số giữa dự đoán vị trí của các bounding box (hộp giới hạn) so với vị trí thực tế của đối tượng. YOLOv8 sử dụng độ đo IoU (Intersection over Union) để đo lường mức độ chồng lấn giữa bounding box dự đoán và bounding box thực tế. Các bounding box có IoU cao hơn ngưỡng được xem là dự đoán chính xác. lặp qua từng cell và từng Bounding Box trong cell, nếu cell đó tồn tại object thì thực hiện tính Loss, còn không thì Loss = 0

$$L_{loc} = \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^B 1_i^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (w_i - \hat{w}_i)^2 + (h_i - \hat{h}_i)^2 \right]$$

Trong đó:

λ_{coord} : là hệ số trọng số cho các phân tử vị trí (trường là 5)

1_i^{obj} : hàm chỉ số để xác định xem bounding box dự đoán i có chứa đối tượng không

(x_i, y_i, w_i, h_i) : các thông số của bounding box dự đoán

$(\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i)$: các thông số của bounding box thực tế.

+ **Classification loss** (hàm mất mát về phân loại): độ lỗi của việc dự đoán loại nhãn của object, hàm lỗi này chỉ tính trên những ô vuông có xuất hiện object, còn những ô vuông khác ta không quan tâm. Nếu cell đó tồn tại object thì thực hiện tính Loss, còn không thì Loss = 0. Classification loss được tính bằng công thức sau:

$$L_{classification} = \sum_{i=0}^{s^2} 1_i^{obj} \sum_{c \in class} (p_i(C) - \hat{p}_i(C))^2$$

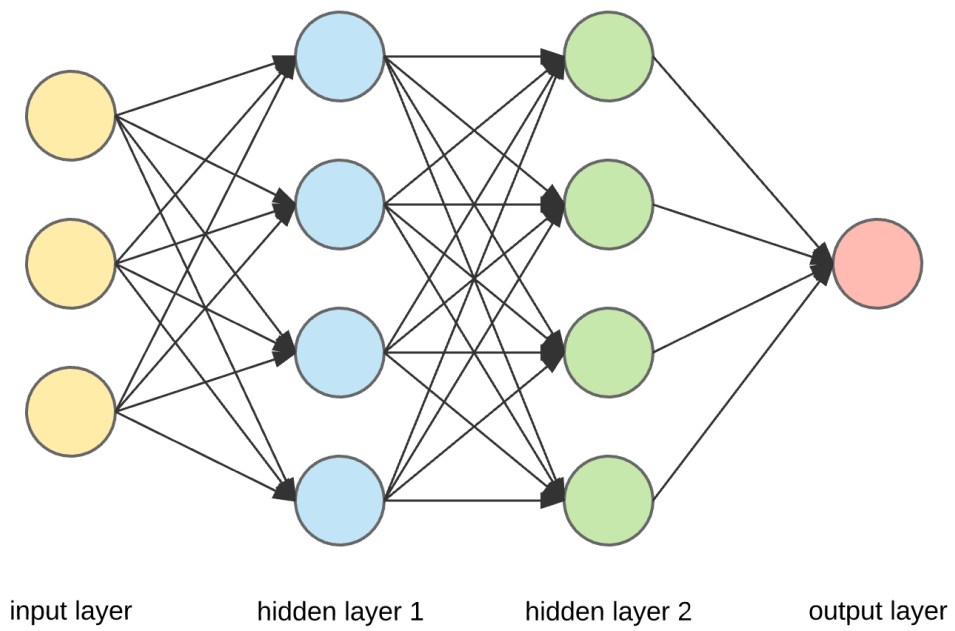
Trong đó:

1_i^{obj} : là hàm chỉ số để chỉ ra xem ô vuông i có chứa đối tượng hay không.

Bảng 1 nếu ô vuông đang xét có Object và ngược lại

$p_i(C)$: xác suất dự đoán cho lớp c trong ô vuông i

$\hat{p}_i(C)$: xác suất thực tế cho lớp c trong ô vuông i



Hình 2.6: Mô hình mạng lưới neural đơn giản
(Nguồn: HackMD. *Mô Hình Nhận Dạng Vật Thể YOLOv8*)

CHƯƠNG 3

THỰC NGHIỆM

3.1. Môi trường thực nghiệm

Về phần cứng thiết bị dùng để huấn luyện mô hình:

Bộ xử lý	
Công nghệ CPU	Core i7 10750H
Tốc độ CPU	2.60 GHz - 5.00 GHz
Số nhân – số luồng	6 nhân – 12 luồng
RAM	
Loại RAM	DDR4 – 16GB
Tốc độ RAM	2933MHz
Card đồ họa	
Model	GeForce GTX 1650
Bộ nhớ	4GB GDDR6
Card onboard	
Model	630
Xung nhịp	350MHz – 1150MHz
Bộ nhớ	
Loại	SSD 512GB

Bảng 3.1: Phần cứng dùng để huấn luyện mô hình

Cài đặt môi trường trên hệ điều hành Windows

	Môi trường
1	Cài đặt Python 3
2	Cập nhật pip
3	Cài đặt PyTorch
4	Cài đặt thư viện bổ sung
5	Cài đặt ultralytics

Bảng 3.2 : Yêu cầu môi trường

Diễn giải bảng:

Để cài đặt và sử dụng YOLOv8, trước hết cần cài đặt Python 3. Python 3 sẽ cung cấp pip để quản lý các thư viện một cách hiệu quả. Tuy nhiên, để đảm bảo sự tương thích và tích hợp hoàn hảo với các phiên bản mới nhất của các thư viện, nên sử dụng phiên bản pip có sẵn mới nhất. Để YOLOv8 hoạt động, cần cài đặt

PyTorch, một framework học máy mạnh mẽ được phát triển bởi Meta AI, cùng với việc cài đặt một số thư viện bổ sung liên quan. Sau khi đã chuẩn bị môi trường với PyTorch, cũng cần cài đặt bộ thư viện ultralytics để đảm bảo các tính năng của YOLOv8 hoạt động một cách đầy đủ và hiệu quả nhất.

3.2. Chuẩn bị dữ liệu

Tập dữ liệu được sử dụng trong báo cáo này đã được lấy từ một tập dữ liệu tổng hợp tại bãi giữ xe của Trường Đại Học Nam Cần Thơ. Tập dữ liệu này đã trải qua nhiều giai đoạn xử lý khác nhau để đảm bảo tính chính xác và hiệu quả của nó. Đầu tiên, các bước tiền xử lý dữ liệu đã được thực hiện cẩn thận để loại bỏ các lỗi và thiếu sót. Sau đó, tập dữ liệu đã được tăng cường thêm thông qua nhiều phương pháp khác nhau nhằm cải thiện chất lượng và tính thực tế của tập dữ liệu. Nguồn của tập dữ liệu này được nhận từ quá trình thu thập và định danh dữ liệu

Tập dữ liệu bao gồm 2 thư mục chứa 2408 ảnh và 2408 label chứa tọa độ đối tượng cần nhận dạng là biển số xe trong ảnh tương ứng. Tuy nhiên để tiến hành train và tùy thuộc vào mô hình sử dụng thì tập dữ liệu cần phải được phân bổ lại theo các tính toán phù hợp với mô hình đó.

3.3. Phân bổ dữ liệu

Để đảm bảo tính khách quan và chính xác của mô hình, tập dữ liệu sẽ được lấy ra 2398 ảnh để thực hiện quá trình train và 10 ảnh cho quá trình test sơ bộ sau khi train chia thành ba phần: tập huấn luyện (train set), tập xác thực (validation set), và tập kiểm tra (test set). Cụ thể:

Tập huấn luyện (Train set):

Chiếm khoảng 70% tổng số dữ liệu 2398 ảnh . Được sử dụng để huấn luyện mô hình. Bao gồm các ảnh và label tương ứng để mô hình học cách nhận diện biển số xe từ dữ liệu đầu vào.

Tập xác thực (Validation set):

Chiếm khoảng 20% tổng số dữ liệu 2398 ảnh. Được sử dụng để điều chỉnh các siêu tham số (hyperparameters) của mô hình. Giúp đánh giá mức độ chính xác của mô hình trong quá trình huấn luyện và ngăn chặn việc quá khớp (overfitting).

Tập kiểm tra (Test set):

Chiếm khoảng 10% tổng số dữ liệu 2398 ảnh. Được sử dụng để đánh giá hiệu năng thực sự của mô hình sau khi quá trình huấn luyện hoàn tất. Cung cấp một cái nhìn tổng quan về khả năng của mô hình khi làm việc với dữ liệu mới và chưa từng thấy trước đó.

Quy trình phân bổ dữ liệu: Quy trình phân bổ dữ liệu có thể được thực hiện như sau:

Chia ngẫu nhiên: Các ảnh và label sẽ được chọn ngẫu nhiên để đảm bảo tính khách quan và đa dạng của các tập dữ liệu.

Bảo đảm sự cân bằng: Đảm bảo rằng mỗi tập dữ liệu đều có sự phân bố đồng đều về các trường hợp khác nhau, ví dụ như các biển số xe ở nhiều góc độ và điều kiện ánh sáng khác nhau.

Lưu trữ và quản lý: Các tập dữ liệu sau khi được phân bổ sẽ được lưu trữ trong các thư mục riêng biệt để tiện cho quá trình huấn luyện, xác thực và kiểm tra mô hình. Cụ thể như sau:

Thư mục	Nội dung	Hình ảnh	Nhãn
train/	Dữ liệu huấn luyện	train/images/	train/labels/
val/	Dữ liệu xác thực	val/ mages/	val/labels/
test/	Dữ liệu kiểm tra	test/images/	test/labels/

Bảng 3.3: Lưu trữ và quản lý các tập dữ liệu

Việc phân bổ 2408 ảnh và 2408 label như sau:

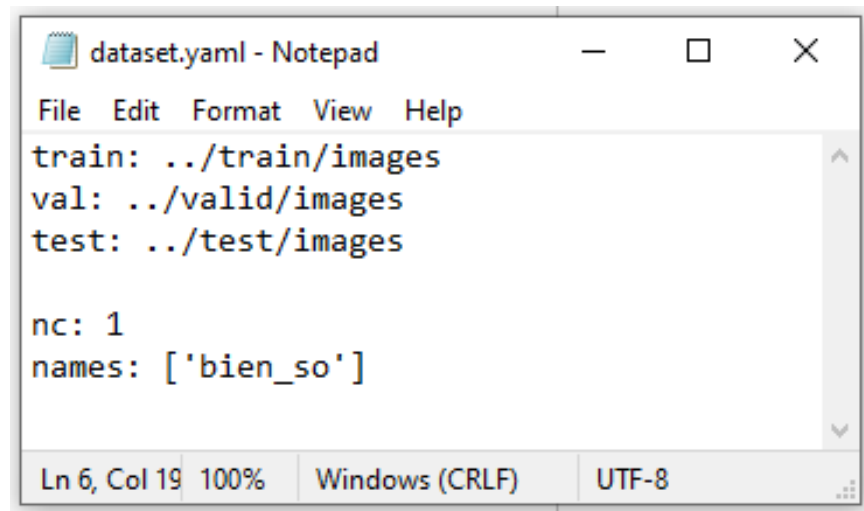
Tập thư mục	Hình ảnh	Tỉ lệ (%)
Tập huấn luyện (Train set)	1679 ảnh và 1679 label	70
Tập xác thực (Validation set)	480 ảnh và 480 label	20
Tập kiểm tra (Test set)	239 ảnh và 239 label	10

Bảng 3.4: Phân bổ ảnh và label

Cấu trúc thư mục để đưa vào quá trình train yolo bao gồm:

- Các thư mục train, val, test
- File dataset.yaml

Nội dung của file dataset.yaml như sau:



```
dataset.yaml - Notepad
File Edit Format View Help
train: ../train/images
val: ../valid/images
test: ../test/images

nc: 1
names: ['bien_so']

Ln 6, Col 19 100% Windows (CRLF) UTF-8
```

Hình 3.1.: Nội dung file dataset.yaml

Ý nghĩa các thành phần trong file dataset.yaml

- Train: ‘../train/images’

Đường dẫn tới thư mục chứa hình ảnh dùng để huấn luyện mô hình.

YOLO sẽ sử dụng các hình ảnh trong thư mục này cùng với các file nhãn tương ứng để học cách nhận diện các đối tượng.

- val: ../valid/images’

Đường dẫn tới thư mục chứa hình ảnh dùng để xác thực mô hình..

Dữ liệu này được dùng để đánh giá hiệu suất của mô hình trong quá trình huấn luyện, giúp tránh overfitting.

- test: ../test/images’

Đường dẫn tới thư mục chứa hình ảnh dùng để kiểm tra mô hình sau khi huấn luyện xong một epochs.

Đây là tập dữ liệu mà mô hình chưa thấy trong quá trình huấn luyện và xác thực, giúp đánh giá khả năng tổng quát hóa của mô hình.

- nc: 1

Số lượng lớp (classes) cần nhận diện. Trong trường hợp này, nc: 1 có nghĩa là chỉ có một lớp duy nhất cần nhận diện.

- names: ['bien_so']

Danh sách các tên lớp. Ở đây có một lớp duy nhất tên là 'bien_so_xe_may' (biển số xe máy).





Mỗi nhãn trong tập dữ liệu sẽ được gán một tên lớp từ danh sách này.

Chức năng của file dataset.yaml:

Định nghĩa cấu trúc tập dữ liệu: File này giúp YOLO biết được nơi lưu trữ hình ảnh cho quá trình huấn luyện, xác thực và kiểm tra.

Định nghĩa các lớp đối tượng: File này cũng xác định số lượng lớp và tên các lớp đối tượng mà mô hình cần nhận diện.

Tự động hóa quy trình huấn luyện: Khi chạy script huấn luyện YOLO, file dataset.yaml sẽ được đọc để lấy thông tin về cấu trúc tập dữ liệu và lớp đối tượng, giúp tự động hóa quá trình chuẩn bị dữ liệu và huấn luyện mô hình.

Name	Date modified	Type	Size
 train	22/06/2024 4:00 AM	File folder	
 test	22/06/2024 4:00 AM	File folder	
 valid	22/06/2024 4:00 AM	File folder	
 dataset.yaml	22/06/2024 3:45 AM	Yaml Source File	1 KB

Hình 3.2: Bộ dữ liệu

Với cấu trúc như trên, file dataset.yaml cung cấp mọi thông tin cần thiết để YOLO có thể bắt đầu huấn luyện mô hình nhận diện biển số xe.

Kết luận

Quá trình phân bổ dữ liệu là một bước quan trọng trong việc chuẩn bị dữ liệu cho mô hình học máy. Bằng cách phân chia dữ liệu thành các tập huấn luyện, xác thực và kiểm tra, chúng ta có thể đảm bảo rằng mô hình được đánh giá một cách toàn diện và chính xác, đồng thời giảm thiểu rủi ro của việc quá khớp. Việc thực hiện đúng quy trình này sẽ giúp nâng cao hiệu quả và độ tin cậy của mô hình nhận diện biển số xe.

3.4. Huấn luyện mô hình

Riêng việc huấn luyện mô hình ta có thể sử dụng thiết bị có card đồ họa hoặc sử dụng Google Colab để thực hiện quá trình train, trong bài báo cáo này em sử dụng phương pháp train trên thiết bị có card đồ họa

Ở phiên bản YOLOv8 đã có cung cấp hai phương thức thao tác với mô hình này thông qua: giao diện dòng lệnh CLI và thông qua code python

Khi train mô hình, thường được gợi ý dùng với giao diện dòng lệnh “CLI” để giảm sự phức tạp của quá trình, giúp chỉ tập trung vào các thông tin quan trọng.

Lệnh CLI để thực thi quá trình train:

```
yolo detect train data=E:\vanhuu\tr\dataset.yaml
```

```
model=yolov8n.yaml epochs=100 imgsz=640
```

Trong đó có các thông số cần phải quan tâm đó là giá trị epochs, imgsz và device.

- epochs: là số lần mà mô hình sẽ lặp qua toàn bộ dữ liệu trong tập học, ví dụ với epochs = 100 thì mô hình sẽ lặp qua 100 lần toàn bộ dữ liệu trong tập học. Epochs càng cao thì tỉ lệ nhận dạng càng được cải thiện

- imgsz: là giá trị để mô hình biết các ảnh có kích thước là bao nhiêu, giá trị này thường được khuyến cáo nên ở 640.

- device: là giá trị để biến mô hình sẽ được hệ thống nào thực thi “CPU, GPU”, thông thường các mô hình học máy, AI nói chung và mô hình yolo sẽ được chạy trên GPU thông qua thư viện Cuda để tối ưu tốc độ học của mô hình.

- data: là giá trị chứa đường dẫn đến tệp chứa tệp yaml, tệp này sẽ chứa các thông tin hướng dẫn mô hình lấy ra các tập học cần thiết.

Tiến hành train dữ liệu:

```

<1> cmd
95/100      2.24G      2.371      3.835      2.945      16      640: 100% | 1/1 [00:01<00:00, 1.08s/it]
           Class      Images      Instances      Box(P      R      mAP50      mAP50-95): 100% | 1/1 [00:00<00:00, 21.34it/s]
           all        2          2          0          0          0          0

Epoch      GPU_mem      box_loss      cls_loss      dfl_loss      Instances      Size
96/100      2.24G      2.107      3.808      2.634      16      640: 100% | 1/1 [00:01<00:00, 1.08s/it]
           Class      Images      Instances      Box(P      R      mAP50      mAP50-95): 100% | 1/1 [00:00<00:00, 15.99it/s]
           all        2          2          0          0          0          0

Epoch      GPU_mem      box_loss      cls_loss      dfl_loss      Instances      Size
97/100      2.127      4.011      2.626      16      640: 100% | 1/1 [00:01<00:00, 1.08s/it]
           Class      Images      Instances      Box(P      R      mAP50      mAP50-95): 100% | 1/1 [00:00<00:00, 21.33it/s]
           all        2          2          0          0          0          0

Epoch      GPU_mem      box_loss      cls_loss      dfl_loss      Instances      Size
98/100      2.24G      2.252      3.85      3.09      16      640: 100% | 1/1 [00:01<00:00, 1.08s/it]
           Class      Images      Instances      Box(P      R      mAP50      mAP50-95): 100% | 1/1 [00:00<00:00, 17.30it/s]
           all        2          2          0          0          0          0

Epoch      GPU_mem      box_loss      cls_loss      dfl_loss      Instances      Size
99/100      2.24G      2.125      3.945      3.004      16      640: 100% | 1/1 [00:01<00:00, 1.07s/it]
           Class      Images      Instances      Box(P      R      mAP50      mAP50-95): 100% | 1/1 [00:00<00:00, 18.31it/s]
           all        2          2          0          0          0          0

Epoch      GPU_mem      box_loss      cls_loss      dfl_loss      Instances      Size
100/100     2.24G      1.991      3.903      2.847      16      640: 100% | 1/1 [00:01<00:00, 1.09s/it]
           Class      Images      Instances      Box(P      R      mAP50      mAP50-95): 100% | 1/1 [00:00<00:00, 18.18it/s]
           all        2          2          0          0          0          0

100 epochs completed in 0.052 hours.
Optimizer stripped from runs\detect\train6\weights\last.pt, 6.2MB
Optimizer stripped from runs\detect\train6\weights\best.pt, 6.2MB

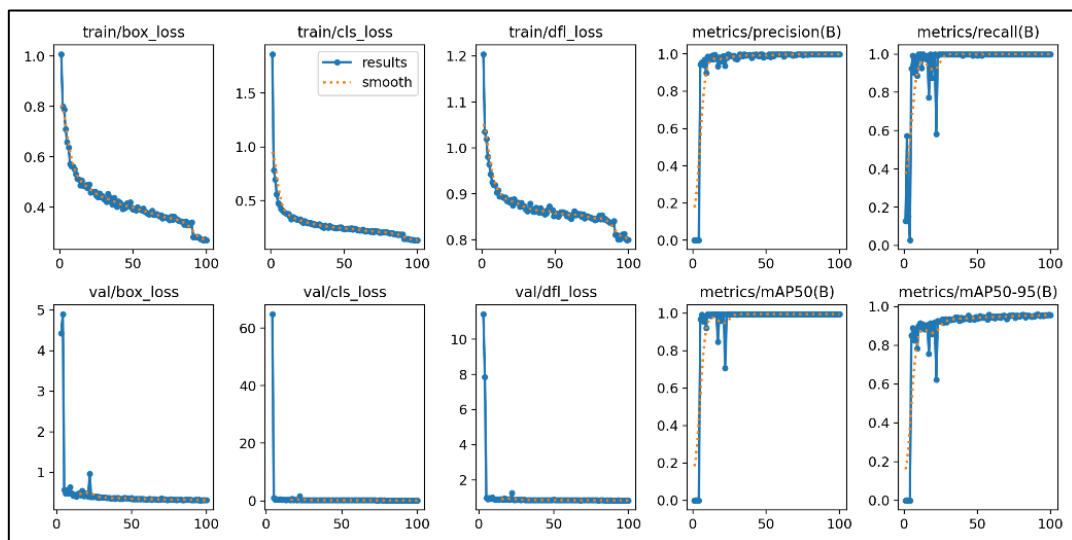
Validating runs\detect\train6\weights\best.pt...
Ultralytics YOLOv8.1.9 Python-3.9.0 torch-2.2.0+cu118 CUDA:0 (NVIDIA GeForce GTX 1650, 4096MiB)
YOLOv8n summary (fused): 168 layers, 3005843 parameters, 0 gradients, 8.1 GFLOPs
           Class      Images      Instances      Box(P      R      mAP50      mAP50-95): 100% | 1/1 [00:00<00:00, 15.94it/s]
           all        2          2          0          0          0          0

Speed: 0.0ms preprocess, 16.7ms inference, 0.0ms loss, 8.5ms postprocess per image
Results saved to
Learn more at https://docs.ultralytics.com/modes/train

(train) E:\vanhuu>

```

Hình 3.4: Tiến hành train dữ liệu



Hình 3.5: Biểu đồ kết quả sau khi train

Sau khi kết thúc quá trình lập lần thứ 100 tương ứng Epochs = 100, mô hình yolo sẽ tiến hành tổng hợp dữ liệu và cho ra kết quả.

Một số định nghĩa:

Mất mát (loss) của các hộp dự đoán (bounding boxes): Đo lường sự khác biệt giữa vị trí/ kích thước hộp dự đoán và hộp thực tế. Giúp mô hình học cách dự đoán chính xác vị trí/kích thước đối tượng.

Mất mát phân loại (classification loss): Đo lường sự khác biệt giữa nhãn dự đoán và nhãn thực tế. Giúp mô hình học cách phân loại đúng đối tượng.

Mất mát của DFL: Điều chỉnh trọng số của các ví dụ huấn luyện dựa trên độ khó của chúng. Giúp mô hình tập trung vào các ví dụ khó, cải thiện khả năng tổng quát hóa.

Độ chính xác (precision) của mô hình: Tỷ lệ dự đoán đúng trong số các dự đoán mô hình cho là đúng. Cho biết mức độ ít đưa ra dự đoán sai của mô hình. Tỷ lệ hợp lý thường trên 90%.

Độ nhạy (recall) của mô hình: Tỷ lệ phát hiện đúng các đối tượng thực sự có trong dữ liệu. Cho biết mức độ ít bỏ sót các đối tượng của mô hình. Tỷ lệ hợp lý: thường trên 90%.

Biểu đồ huấn luyện (training):

Biểu đồ thứ 1: train/box_loss

- Biểu đồ này thể hiện sự mất mát (loss) của các hộp dự đoán (bounding boxes) trong quá trình huấn luyện. Giá trị mất mát giảm dần theo số epoch từ 1.0 xuống dưới 0.4 cho thấy mô hình đang cải thiện khả năng định vị các đối tượng.

Biểu đồ thứ 2: train/cls_loss

- Biểu đồ này thể hiện sự mất mát của việc phân loại (classification loss) trong quá trình huấn luyện. Giá trị giảm dần từ trên 1.5 xuống dưới 0.5 cho thấy mô hình ngày càng tốt hơn trong việc phân loại các đối tượng.

Biểu đồ thứ 3: train/df1_loss

- Biểu đồ này thể hiện sự mất mát của DFL trong quá trình huấn luyện. Sự giảm dần từ 1.2 xuống mức dưới 0.8 của giá trị này cho thấy mô hình đang ngày càng tốt hơn trong việc học dự đoán vị trí đối tượng chính xác hơn.

Biểu đồ đánh giá (validation):

Biểu đồ thứ 4: metrics/precision(B)

- Biểu đồ này thể hiện độ chính xác cao trong việc dự đoán đối tượng. Với giá trị ở lần lặp đầu tiên đến lần lặp thứ 100 là 0.0 – 1.0

Biểu đồ thứ 5: metrics/recall(B)

- Biểu đồ này thể hiện sự mất mát của việc phân loại trong quá trình đánh giá. Mặc dù có một số biến động, giá trị này cũng giảm dần, cho thấy khả năng phân loại của mô hình đang cải thiện.

Biểu đồ thứ 6: val/box_loss

- Biểu đồ này học cách khái quát hóa tốt hơn trong việc xác định đối tượng. Giá trị giảm dần từ 5 (ở epoch thứ 1) xuống dưới 1 (epoch thứ 100) chỉ ra rằng mô hình đang học tốt hơn phân phối của các đối tượng trên dữ liệu kiểm tra.

Các chỉ số đánh giá (Evaluation Metrics)

Biểu đồ thứ 7: val/cls_loss

- Biểu đồ này thể hiện độ chính xác (precision) của mô hình. Giá trị duy trì ở mức cao và ổn định, cho thấy mô hình có tỷ lệ chính xác cao trong việc xác định đối tượng.

Biểu đồ thứ 8: val/df1_loss

- Biểu đồ này thể hiện độ nhạy (recall) của mô hình. Giá trị duy trì cao và ổn định, cho thấy mô hình có khả năng phát hiện hầu hết các đối tượng trong dữ liệu.

Biểu đồ thứ 9: metrics/mAP50(B)

- Biểu đồ này thể hiện mAP (mean Average Precision) với ngưỡng IoU là 0.5. Giá trị này cao và ổn định, chỉ ra rằng mô hình có hiệu suất tốt trong việc xác định đối tượng với mức độ chính xác cao.

Biểu đồ thứ 10: metrics/mAP50-95(B)

- Biểu đồ này thể hiện mAP với các ngưỡng IoU từ 0.5 đến 0.95. Giá trị này cũng cao và ổn định, cho thấy mô hình duy trì hiệu suất tốt trên nhiều ngưỡng khác nhau của IoU.

Nhìn chung, các biểu đồ tổn thất (loss) cho thấy mô hình đã học được tốt từ dữ liệu huấn luyện khi tổn thất giảm dần theo số lượng epoch. Các biểu đồ độ chính xác, độ hồi phục và mAP cho thấy mô hình có hiệu suất tốt trong việc phát hiện và phân loại đối tượng. Các biểu đồ này đều ổn định sau một số lượng epoch nhất định, điều này cho thấy mô hình đã đạt đến mức hiệu suất tối ưu và không bị overfitting.

Sau quá trình huấn luyện, mô hình tạo ra một tập hợp các kết quả bao gồm hai tệp quan trọng: `best.pt` và `last.pt`. Trong đó:

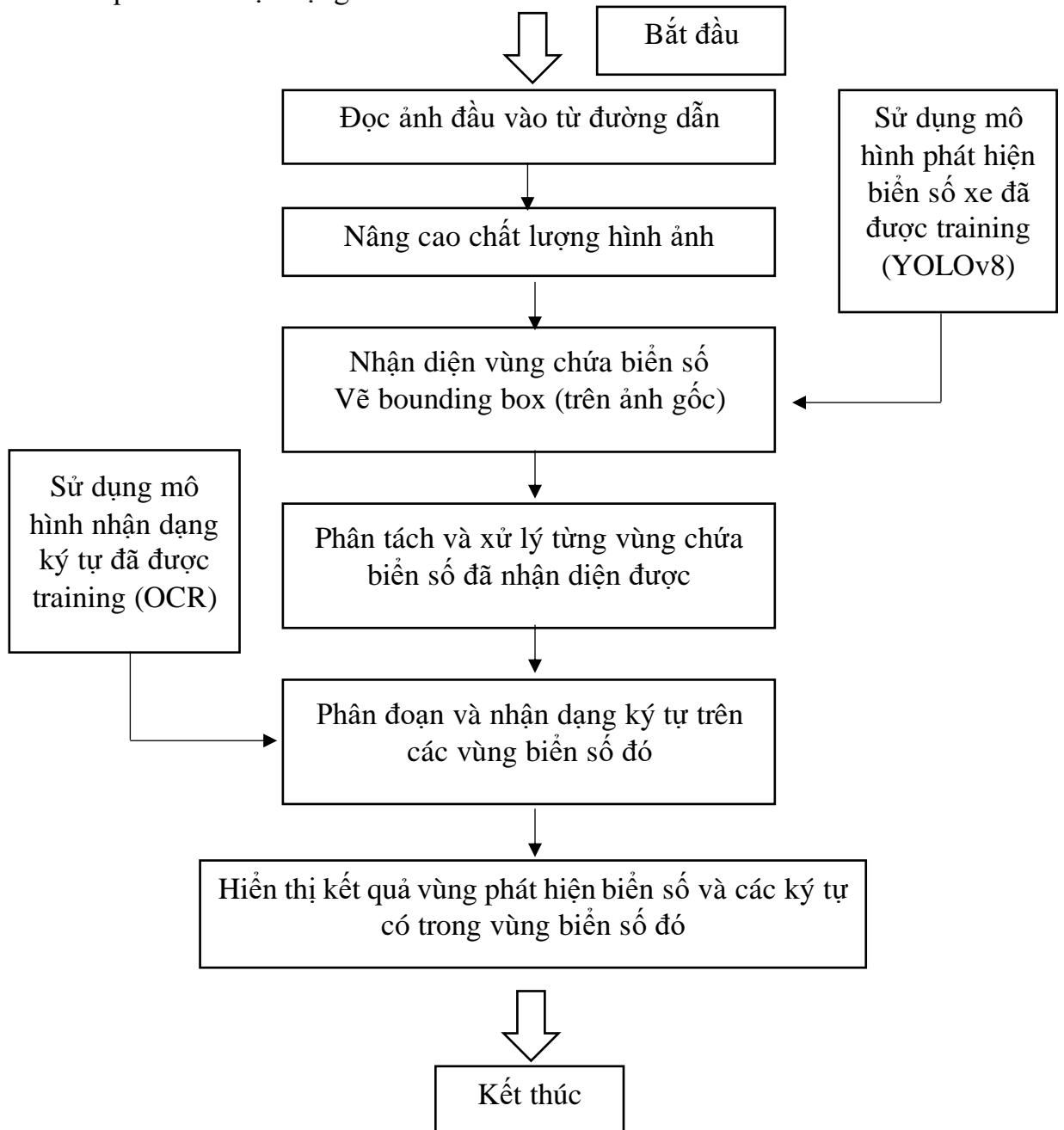
- **best.pt**: Tệp này lưu trữ các trọng số tại điểm mà mô hình đạt hiệu suất tốt nhất trong quá trình huấn luyện. Khi triển khai mô hình vào ứng dụng thực tế, tệp `best.pt` thường được sử dụng để đảm bảo mô hình hoạt động với độ chính xác cao nhất.

- **last.pt**: Tệp này lưu trữ các trọng số từ epoch đầu tiên đến epoch cuối cùng của quá trình huấn luyện. Tệp `last.pt` thường được sử dụng khi cần tiếp tục huấn luyện mô hình với dữ liệu mới, giúp cập nhật và cải thiện khả năng nhận diện của mô hình theo thời gian.

Sử dụng tệp `best.pt` trong các ứng dụng thực tế giúp đảm bảo hiệu suất tối ưu, trong khi tệp `last.pt` cung cấp sự linh hoạt để mô hình có thể được cải thiện liên tục.

3.5 Thử nghiệm và đánh giá.

Lưu đồ quá trình nhận dạng biển số xe



Hình 3.6 : Lưu đồ chi tiết của hệ thống nhận dạng biển số xe

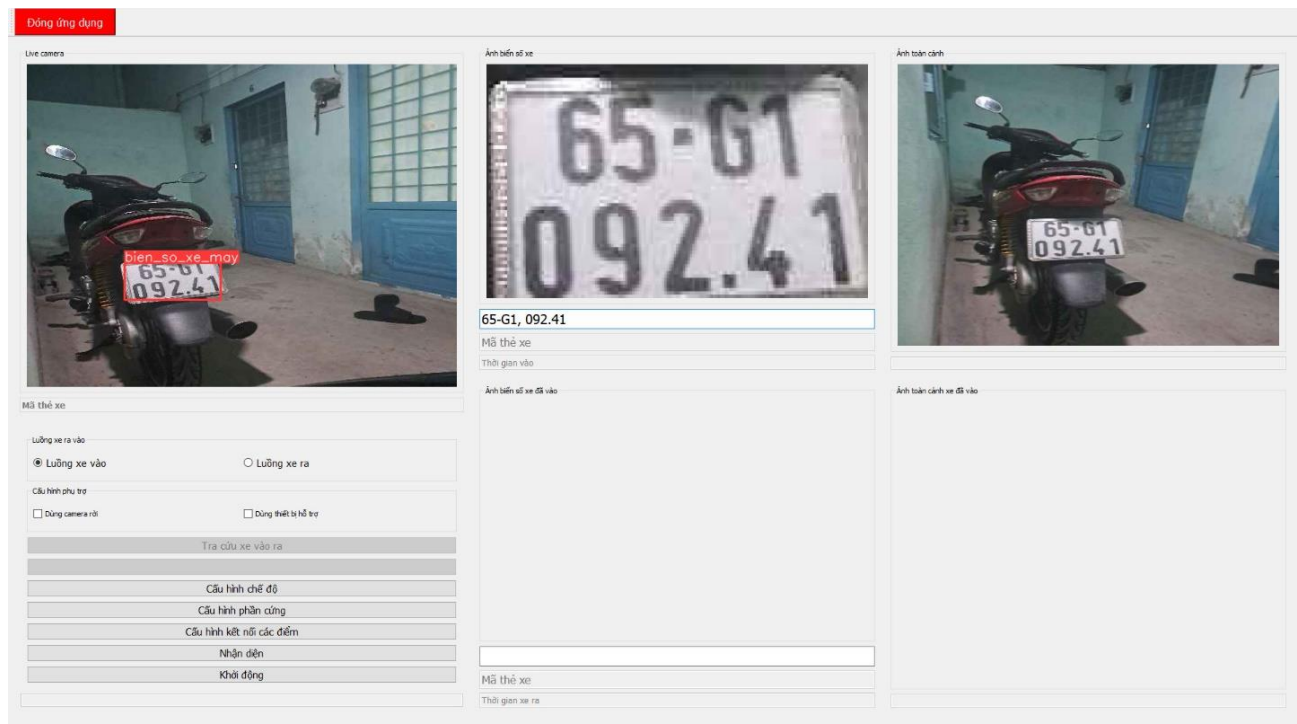
3.5.1. Thử nghiệm chương trình

Ảnh đầu vào là camera của điện thoại

YOLO phát hiện vị trí của biển số và vẽ 1 bounding box lên ảnh

CV2 dựa theo bounding box đã vẽ và tiến hành cắt biển số

EasyOCR tiến hành nhận dạng kí tự trong biển số và hiển thị ở textbox (đầu ra)



Hình 3.7 Thử nghiệm chương trình nhận dạng

3.5.1. Đánh giá chương trình

Chương trình thử nghiệm đã cho thấy hiệu quả rõ rệt trong việc phát hiện và nhận dạng biển số xe. Các bước xử lý của YOLO, CV2 và EasyOCR đã phối hợp một cách nhịp nhàng, tạo nên một hệ thống nhận diện biển số xe chính xác và nhanh chóng. YOLO đã thực hiện tốt nhiệm vụ phát hiện vị trí biển số với độ chính xác cao. kết quả này khẳng định rằng mô hình đã được huấn luyện tốt và có khả năng ứng dụng thực tế cao trong việc nhận dạng biển số xe, mở ra tiềm năng lớn cho các ứng dụng giám sát, quản lý giao thông và các lĩnh vực liên quan. Việc sử dụng các tệp trọng số best.pt đảm bảo rằng mô hình đạt được hiệu suất tối ưu, trong khi tệp last.pt cung cấp khả năng linh hoạt để tiếp tục cải thiện và cập nhật mô hình.

CHƯƠNG 4

KẾT QUẢ NGHIÊN CỨU

4.1. Kết quả huấn luyện và kiểm tra mô hình

Trong quá trình huấn luyện mô hình nhận dạng biển số xe, em đã sử dụng YOLOv8 kết hợp với EasyOCR để đạt hiệu suất tối ưu. Dữ liệu huấn luyện được chuẩn bị kỹ lưỡng từ bãi giữ xe của Trường Đại Học Nam Cần Thơ và mô hình đã trải qua nhiều lần lặp (epochs) để đảm bảo hiệu suất tối đa. Kết quả cho thấy mô hình YOLOv8 học tốt từ dữ liệu huấn luyện, với các chỉ số tổn thất giảm dần và độ chính xác tăng lên qua các epoch. Khi áp dụng trên tập dữ liệu xác thực, mô hình tiếp tục thể hiện khả năng ổn định và hiệu quả, với độ chính xác và độ hồi phục tăng lên, chứng tỏ khả năng nhận diện đối tượng một cách chính xác và toàn diện. Khi kiểm tra trên tập dữ liệu kiểm tra, các kết quả đạt được rất khả quan, cho thấy mô hình có khả năng tổng quát hóa tốt và không bị quá khớp (overfitting).

Để tăng cường khả năng nhận dạng và phân loại các ký tự trên biển số xe, em đã tích hợp EasyOCR vào hệ thống. Sau khi YOLOv8 xác định được vùng chứa biển số xe trong ảnh, EasyOCR được sử dụng để trích xuất các ký tự từ vùng đó. EasyOCR, một công cụ OCR mã nguồn mở, giúp chuyển đổi hình ảnh thành văn bản một cách dễ dàng và hiệu quả. Việc kết hợp EasyOCR vào hệ thống đã cải thiện đáng kể độ chính xác và hiệu suất tổng thể, giúp hệ thống nhận diện các biển số xe một cách toàn diện hơn, ngay cả khi các ký tự trên biển số không rõ ràng hoặc bị che khuất một phần.

Nhìn chung, các kết quả từ quá trình huấn luyện, xác thực và kiểm tra cho thấy mô hình kết hợp YOLOv8 và EasyOCR đã thành công trong việc nhận dạng biển số xe với độ chính xác và hiệu suất cao. Hệ thống đã chứng minh được tính linh hoạt và khả năng nhận diện các biển số xe trong nhiều điều kiện khác nhau, đảm bảo hiệu quả và độ tin cậy cao..

4.2. Phân tích kết quả

Kết quả nghiên cứu được phân tích để hiểu các yếu tố ảnh hưởng đến hiệu suất của mô hình YOLOv8 và EasyOCR trong nhận dạng biển số xe. Độ phức tạp của

dữ liệu đầu vào đóng vai trò quan trọng đối với hiệu suất nhận dạng. Dữ liệu có độ phức tạp cao như biển số bị che khuất, mờ, nhiễu, hoặc bị che chắn bởi các đối tượng khác có thể làm giảm độ chính xác của mô hình. Việc sử dụng các kỹ thuật tiền xử lý dữ liệu như làm sạch và làm rõ hình ảnh có thể giúp cải thiện hiệu suất nhận dạng.

Sự ảnh hưởng của môi trường cũng đáng kể, bao gồm ánh sáng chói làm mất chi tiết và giảm độ tương phản của biển số, biển số bẩn làm mờ hoặc che khuất các ký tự, khoảng cách quá xa làm giảm độ phân giải của biển số trên hình ảnh, và góc chụp quá nghiêng gây biến dạng hình dáng của biển số, làm khó khăn cho việc xác định vị trí và phân loại các ký tự.

Việc tích hợp EasyOCR vào hệ thống nhận dạng giúp tăng cường khả năng nhận dạng và phân loại các ký tự trên biển số xe. EasyOCR, một công cụ OCR mã nguồn mở, giúp chuyển đổi hình ảnh thành văn bản một cách dễ dàng và hiệu quả. Tuy nhiên, EasyOCR có thể gặp khó khăn trong việc nhận diện một số ký tự, dẫn đến nhầm lẫn giữa các ký tự tương tự nhau, chẳng hạn như "1" có thể bị nhầm là "7" và "G" có thể bị nhầm là "6".

Nhìn chung, kết quả nghiên cứu khẳng định rằng việc áp dụng công nghệ nhận dạng biển số xe bằng thị giác máy tính là một giải pháp hiệu quả và khả thi cho các nhu cầu quản lý hiện đại. Bộ dữ liệu và mô hình đã huấn luyện được lưu trên GitHub, cho phép các sinh viên hoặc nghiên cứu sinh tiếp cận và phát triển thêm. Sự kết hợp giữa YOLOv8 và EasyOCR không chỉ cải thiện hiệu suất nhận dạng mà còn đảm bảo độ tin cậy cao trong nhiều điều kiện khác nhau.

CHƯƠNG 5

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

Trong nghiên cứu này, em đã tiến hành một cuộc thử nghiệm toàn diện về việc áp dụng phương pháp nhận diện biển số xe máy sử dụng mô hình YOLO kết hợp với EasyOCR. Mục tiêu của em là có thể nghiên cứu và chứng thực công năng cùng hiệu suất của một hệ thống nhận diện biển số xe máy tự động, hiệu quả và đáng tin cậy.

Kết quả của cuộc thử nghiệm cho thấy rằng hệ thống nhận diện rất hiệu quả đối với các đối tượng trong ảnh và video. Tuy nhiên, về phần nhận diện ký tự, vẫn còn tồn tại nhiều sai sót và cần cải thiện thêm. Cụ thể, EasyOCR đôi khi nhầm lẫn giữa các ký tự tương tự như "1" và "7", "G" và "6". Những sai sót này đòi hỏi các giải pháp tiền xử lý dữ liệu mạnh mẽ hơn hoặc cải thiện thuật toán nhận diện để tăng độ chính xác.

Tóm lại, nghiên cứu của em đã chứng minh tính khả thi của phương pháp nhận diện biển số xe máy sử dụng mô hình YOLO kết hợp với EasyOCR. Kết quả này có thể cung cấp cơ sở cho việc triển khai hệ thống nhận diện biển số xe máy tự động trong các ứng dụng thực tế như quản lý bãi đậu xe.

Qua quá trình thực hiện nghiên cứu, các mục tiêu ban đầu đã được hoàn thành. Các công việc đã thực hiện bao gồm thu thập bộ dữ liệu đa dạng và đầy đủ các biến thể cần thiết, áp sát với điều kiện hoạt động thực tế. Nghiên cứu đã đánh giá và đưa ra được ý kiến về hệ thống nhận diện biển số bằng YOLO và EasyOCR, khẳng định rằng đây là một lựa chọn tối ưu cho phương pháp học sâu trong hệ thống hiện tại. Cuối cùng, em cũng đã thực hiện thử nghiệm trên bộ ảnh thực tế được chụp từ nhà xe Đại Học Nam Cần Thơ, từ đó đánh giá lại hiệu quả, tính chính xác cũng như hiệu suất của hệ thống.

Nhìn chung, nghiên cứu này đã chứng minh rằng việc áp dụng công nghệ thị giác máy tính với sự kết hợp của YOLO và EasyOCR là một giải pháp khả thi và hiệu quả cho việc nhận diện biển số xe máy trong các điều kiện thực tế khác nhau.

5.2. Hướng phát triển của đề tài

Để nâng cao hiệu quả và mở rộng ứng dụng của mô hình nhận diện biển số xe sử dụng YOLOv8, em đề xuất các hướng phát triển tiếp theo:

Nghiên cứu và áp dụng các phương pháp tiên tiến trong xử lý ảnh: Cải thiện độ chính xác nhận diện ký tự: Tiếp tục nghiên cứu và áp dụng các kỹ thuật xử lý ảnh tiên tiến nhằm cải thiện khả năng nhận diện và phân tích ký tự trên biển số xe. Các phương pháp có thể bao gồm sử dụng mạng nơ-ron tích chập (CNN) đặc thù cho ký tự và kỹ thuật tăng cường học tập để giảm thiểu lỗi nhận diện.

Tối ưu hóa mô hình và triển khai trên các nền tảng nhúng: Tăng cường tốc độ xử lý: Nghiên cứu và tối ưu hóa mô hình YOLOv8 để đảm bảo tốc độ xử lý cao, đặc biệt là khi triển khai trên các nền tảng nhúng như Raspberry Pi, Jetson Nano, hay các hệ thống nhúng khác. Việc này giúp mở rộng khả năng ứng dụng của hệ thống trong các điều kiện thực tế.

Mở rộng ứng dụng vào các lĩnh vực khác: Nghiên cứu và phát triển các ứng dụng mở rộng của mô hình nhận diện biển số xe trong các lĩnh vực như an ninh, quản lý đô thị và giao thông vận tải.

TÀI LIỆU THAM KHẢO

- [1]. Nguyễn Chiến Thắng (2023). *Thử nghiệm YOLOv8 và train với dữ liệu cá nhân*. Mì AI. Truy cập vào 1/3/2024 từ: <https://www.miai.vn/2023/05/19/thu-nghich-yolov8-va-train-voi-du-lieu-ca-nhan-mi-ai/>
- [2]. glenn-jocher (2024). *Model Validation with Ultralytics YOLO*. Ultralytics YOLO Docs. Truy cập vào 2/3/2024 từ: <https://docs.ultralytics.com/modes/val/>
- [3]. Đỗ Gia Huy (2023). *Mô Hình Nhận Dạng Vật Thể YOLOv8*. HackMD. Truy cập ngày: 26/3/2024. Nguồn: <https://hackmd.io/@58ZC49ZfS86wYX--LRGGOg/Viewperm#C-Qu%C3%A1-tr%C3%ACnh-hu%E1%BA%A5n-luy%E1%BB%87n>
- [4]. Việt Hoàng (2019). *Tìm hiểu về YOLO trong bài toán real-time object detection*. Viblo. Truy cập ngày: 26/3/2024. Nguồn: <https://viblo.asia/p/tim-hieu-ve-yolo-trong-bai-toan-real-time-object-detection-yMnKMdvr57P>
- [5]. Hao Ng (2022). *YOLOv1 - Nhận diện vật thể chỉ ngó 1 lần - Khởi nguồn*. Viblo. Truy cập ngày: 26/3/2024. Nguồn: <https://viblo.asia/p/yolov1-nhan-dien-vat-the-chi-ngo-1-lan-khoi-nguon-aNj4v0b0L6r>
- [6]. VBD (2023). *YOLOv8 có gì nâng cấp so với các phiên bản trước?*. VinBigData. Truy cập ngày: 2/3/2024. Nguồn: <https://vinbigdata.com/cong-nghe-hinh-anh/yolov8-co-gi-nang-cap-so-voi-cac-phien-ban-truoc.html>