

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA ĐIỆN TỬ - VIỄN THÔNG**

NHÓM 4

**ỨNG DỤNG MÔ HÌNH MẠNG YOLOV8
TRONG VIỆC NHẬN DẠNG CÁC BIỂN BÁO GIAO THÔNG
Ở VIỆT NAM**

BÁO CÁO ĐỒ ÁN MÔN HỌC

MÔN HỌC : NHẬP MÔN TRÍ TUỆ NHÂN TẠO

Thành phố Hồ Chí Minh – Năm 2024

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA ĐIỆN TỬ - VIỄN THÔNG**

BÁO CÁO ĐỒ ÁN MÔN HỌC

MÔN HỌC: NHẬP MÔN TRÍ TUỆ NHÂN TẠO

**ỨNG DỤNG MÔ HÌNH MẠNG YOLOV8 TRONG VIỆC
NHẬN DẠNG CÁC BIỂN BÁO GIAO THÔNG
Ở VIỆT NAM**

Nhóm sinh viên thực hiện: Nhóm 4

Nguyễn Dũng	MSSV: 20200121
Trần Quang Đài	MSSV: 20200152
Đỗ Quang Hiếu	MSSV: 20200196
Đặng Ngọc Hoàng	MSSV: 20200204
Trần Bích Ngọc	MSSV: 20200280
Lạc Hải Sơn	MSSV: 20200328
Trần Thuận Thiên	MSSV: 20200090

GIẢNG VIÊN GIẢNG DẠY MÔN HỌC:

Ths. Nguyễn Thái Công Nghĩa

Thành phố Hồ Chí Minh – Năm 2024

[illegible]

Giảng viên Nhập môn trí tuệ nhân tạo

Nguyễn Thái Công Nghĩa

LỜI CẢM ƠN

Trước hết, chúng em xin bày tỏ lòng biết ơn sâu sắc nhất đến Thầy ThS. Nguyễn Thái Công Nghĩa, người đã tận tình hướng dẫn, cung cấp kiến thức quý báu cũng như những lời khuyên bổ ích trong suốt quá trình chúng em thực hiện đề tài này. Thầy không chỉ là người hướng dẫn, mà còn là nguồn cảm hứng để chúng em tiếp tục nỗ lực và đổi mới trong suốt quá trình nghiên cứu và phát triển đề tài của chúng em.

Thầy đã luôn kiên nhẫn và dành thời gian để giải đáp mọi thắc mắc, đồng thời đưa ra những chỉ dẫn chi tiết và cần thiết giúp chúng em vượt qua nhiều khó khăn trong quá trình nghiên cứu làm đề tài. Lời khuyên và sự chỉ bảo của Thầy đã vô cùng quan trọng, góp phần không nhỏ vào thành công và sự phát triển của đề tài này.

Do kiến thức và khả năng lý luận còn nhiều hạn chế nên đề tài vẫn còn những thiếu sót nhất định. Em rất mong nhận được những đóng góp của các thầy giáo, cô giáo để đề án môn học của chúng em được hoàn thiện hơn.

Lời cuối cùng, chúng em xin gửi lời cảm ơn chân thành nhất đến Thầy, với lòng biết ơn sâu sắc và tình cảm trân trọng vì sự giúp đỡ và đóng góp không thể thiếu trong quá trình thực hiện đề án của chúng em. Sự hỗ trợ của Thầy đã góp phần quan trọng của đề án và trải nghiệm học tập của chúng em.

Xin cảm ơn Thầy.

TP. Hồ Chí Minh, ngày 10 tháng 06 năm 2024

Ký tên Trưởng nhóm

Dũng

Nguyễn Dũng

TÓM TẮT

Hiện nay, tình trạng giao thông ngày càng phức tạp và số lượng phương tiện gia tăng nhanh chóng, việc nhận dạng và xử lý chính xác các biển báo giao thông trở nên cực kỳ quan trọng để đảm bảo an toàn giao thông. Hệ thống nhận dạng biển báo giao thông tự động không chỉ hỗ trợ người lái xe tuân thủ quy tắc mà còn là một thành phần quan trọng trong việc phát triển các phương tiện tự hành. Mô hình mạng YOLO (You Only Look Once) được chọn để nghiên cứu và ứng dụng trong đề tài này nhờ vào khả năng phát hiện đối tượng nhanh và chính xác trong ảnh.

YOLO là một trong những mô hình deep learning tiên tiến trong lĩnh vực thị giác máy tính. Mô hình này nổi bật nhờ vào khả năng thực hiện phát hiện đối tượng trong một lần nhìn (one look) duy nhất, điều này giúp tăng tốc độ xử lý và giảm thời gian tính toán. Đề tài “Ứng dụng mô hình mạng YOLO trong việc nhận dạng các biển báo giao thông trên đường ở Việt Nam” tập trung vào việc áp dụng công nghệ nhận dạng hình ảnh tiên tiến để phát hiện và nhận diện các biển báo giao thông. Mô hình YOLO, với khả năng nhận dạng đối tượng nhanh và chính xác trong thời gian thực, được chọn để giải quyết vấn đề này. Đề tài bao gồm các bước thu thập và xử lý dữ liệu, huấn luyện mô hình, và đánh giá hiệu quả của hệ thống trong các điều kiện giao thông thực tế tại Việt Nam, hứa hẹn mang lại nhiều lợi ích thiết thực cho việc quản lý và điều tiết giao thông.

Trong bối cảnh này, việc nghiên cứu và ứng dụng mô hình YOLO để nhận dạng các biển báo giao thông ở Việt Nam là một bước tiến quan trọng, mở ra nhiều cơ hội cho việc phát triển các hệ thống giao thông thông minh và an toàn hơn.

MỤC LỤC

LỜI CẢM ƠN	4
TÓM TẮT	1
MỤC LỤC	2
DANH SÁCH CHỮ VIẾT TẮT	4
DANH SÁCH CÁC HÌNH	5
DANH SÁCH CÁC BẢNG.....	8
CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI	9
1.1. BỐI CẢNH ĐỀ TÀI:	9
1.2. LÝ DO THỰC HIỆN ĐỀ TÀI:	9
1.3. MỤC TIÊU CỦA ĐỀ TÀI:.....	9
1.4. CẤU TRÚC ĐỀ TÀI:	10
CHƯƠNG 2: TỔNG QUAN VỀ HỌC SÂU VÀ MÔ HÌNH MẠNG YOLO	11
2.1. TỔNG QUAN VỀ HỌC SÂU:.....	11
2.1.1. Mạng Neural:	11
2.1.2. Mạng Neural lan truyền ngược:.....	12
2.1.3. Một số hàm kích hoạt (Activation functions) phổ biến trong học sâu:	14
2.2. BÀI TOÁN VỀ NHẬN DẠNG ĐỐI TƯỢNG SỬ DỤNG YOLOV8:	18
2.2.1. Giới thiệu YOLO:.....	18
2.2.2. Tổng quan về mô hình YOLOv8:.....	21
2.2.3. Khái quát về hệ thống nhận dạng đối tượng:.....	23
2.2.4. Hệ thống nhận dạng đối tượng sử dụng YOLOv8:	24
2.2.5. Một số khái niệm cơ bản liên quan tới bài toán nhận dạng đối tượng:	25
2.2.6. Một số tham số đánh giá hiệu suất mô hình:	29
2.3. GOOGLE COLAB, KAGGLE, ROBOFLOW VÀ ULTRALYTICS:	32
2.3.1. Google Colab:	32
2.3.2. Kaggle:.....	34
2.3.3. Roboflow:	35
2.3.4. Ultralytics	37
CHƯƠNG 3: TRIỂN KHAI ĐỀ TÀI.....	38
3.1. SƠ ĐỒ KHỐI QUÁ TRÌNH TRIỂN KHAI:.....	38
3.1.1. Quá trình thu thập, gán nhãn và xử lý dữ liệu:	38

3.1.2. Quá trình huấn luyện mô hình:	43
CHƯƠNG 4: ĐÁNH GIÁ KẾT QUẢ VÀ KIỂM THỬ MÔ HÌNH.	45
4.1. KẾT QUẢ SAU KHI HUẤN LUYỆN MÔ HÌNH:	45
4.2. TIẾN HÀNH CHẠY THỬ MÔ HÌNH SỬ DỤNG WEBCAM:	51
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	54
5.1. KẾT LUẬN:	54
5.2. HƯỚNG PHÁT TRIỂN:	54
TÀI LIỆU THAM KHẢO	56

DANH SÁCH CHỮ VIẾT TẮT

STT	Ký hiệu, chữ viết tắt	Chữ viết đầy đủ
1	AI	Artificial Intelligence
2	CNN	Convolutional Neural Network
3	ELU	Exponential Linear Unit
4	GELAN	Generalized Efficient Layer Aggregation Network
5	IoU	Intersection over Union
6	NMS	Non-Maximum Suppression
7	PGI	Programmable Gradient Information
8	ReLU	Rectified Linear Unit
9	YOLO	You Only Look Once

DANH SÁCH CÁC HÌNH

Hình 2-1: Minh họa cho mạng Neural Network	11
Hình 2-2: Kiến trúc mạng Neural	12
Hình 2-3: Mạng Neural lan truyền ngược.....	14
Hình 2-4: Hàm Sigmoid.....	15
Hình 2-5: Hàm Tanh.	15
Hình 2-6: Hàm ReLU.....	16
Hình 2-7: Hàm Leaking ReLU	17
Hình 2-8: Hình ELU.	17
Hình 2-9: Sơ đồ các lớp của mô hình YOLO cơ bản	18
Hình 2-10: Lịch sử phát triển của YOLO	18
Hình 2-11: Sơ đồ kiến trúc mạng YOLO.....	20
Hình 2-12: Kiến trúc mạng chung của mô hình mạng YOLO	21
Hình 2-13: So sánh về độ chính xác của YOLOv8 so với các phiên bản trước đó. ^[6]	22
Hình 2-14: Minh họa cho hệ thống nhận dạng đối tượng.....	24
Hình 2-15: Minh họa cho các hoạt động của YOLO trong nhận dạng đối tượng.	25
Hình 2-16: Minh họa cho hợp giới hạn.....	26
Hình 2-17: Minh họa hộp neo trước.	27
Hình 2-18: Các bản đồ đặc trưng cho hình.....	27
Hình 2-19: Từ 3 bounding box ban đầu cùng bao quanh chiếc xe đã giảm xuống còn một bounding box cuối cùng	28
Hình 2-20: Hình minh họa cho độ chính xác.....	29

Hình 2-21: Hình minh họa cho độ chuẩn xác	30
Hình 2-22: Minh họa cho xác định các bounding box.....	31
Hình 2-23: Công thức tính IOU	32
Hình 2-24: Các mức độ đánh giá sử dụng IOU.	32
Hình 2-25: Logo của Google Colab.....	33
Hình 2-26: Giao diện làm việc của Google Colab.....	34
Hình 2-27: Logo của Kaggle	34
Hình 2-28: Giao diện làm việc của Kaggle.....	35
Hình 2-29: Logo của Roboflow.	36
Hình 2-30: Giao diện làm việc của Roboflow.	36
Hình 2-31: Logo của Ulytralitics	37
Hình 3-1: Một số hình ảnh chụp các biển báo từ nhóm.....	38
Hình 3-2: Minh họa cho các tham số nhãn YOLO	40
Hình 3-3: Quá trình dán nhãn cho các biển báo.	41
Hình 3-4: Cấu trúc file nhãn của YOLO.....	41
Hình 3-5: Điều chỉnh các loại file nhãn.....	42
Hình 3-6: Tập dữ liệu biển báo giao thông.	42
Hình 3-7: Quá trình huấn luyện mô hình với 150 epochs.	44
Hình 4-1: Kết quả sau khi huấn luyện mô hình với số epoch là 100.....	45
Hình 4-2: Confusion matrix.....	46
Hình 4-3: Đường cong F1-Confidence Curve.	46
Hình 4-4: Đường cong Recall-Confidence Curve	47
Hình 4-5: Đường cong Precision-Confidence Curve.....	47

Hình 4-6: Đường cong Precision-Recall Curve.....	48
Hình 4-7: Hình minh họa cho việc dự đoán mô hình	50
Hình 4-8: Một ví dụ khác trong chạy thử mô hình.	51
Hình 4-9: Kết quả chạy mô phỏng đoạn mã.	53

DANH SÁCH CÁC BẢNG

Bảng 2-1: So sánh hiệu suất của các phiên bản YOLOv8.....	23
Bảng 3-1: Tổng hợp 59 loại biển báo thông dụng và mã số của các loại biển báo. 39	
Bảng 4-1: Bảng kết quả xét trên tập validation	49

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

1.1. BỐI CẢNH ĐỀ TÀI:

Hiện nay, Việt Nam đang trải qua giai đoạn phát triển mạnh mẽ về kinh tế và xã hội kéo theo đó là sự gia tăng đáng kể người tham gia phương tiện giao thông. Cùng với sự phát triển này, tình hình giao thông ở Việt Nam ngày càng phức tạp điều này đặt ra những thách thức lớn về an toàn giao thông, đặc biệt là trong việc tuân thủ các quy tắc và biển báo giao thông. Vì vậy, với khả năng nhận dạng nhanh và chính xác, YOLO có thể được ứng dụng rộng rãi trong các hệ thống giám sát giao thông, hỗ trợ lái xe tại Việt Nam.

1.2. LÝ DO THỰC HIỆN ĐỀ TÀI:

Xuất phát từ nhu cầu cấp thiết trong việc nâng cao an toàn giao thông, đáp ứng sự gia tăng nhanh chóng của số lượng phương tiện và tình trạng kẹt xe, tai nạn giao thông ngày càng cao. Công nghệ AI và Deep Learning, đặc biệt là mô hình YOLO mang lại khả năng xử lý dữ liệu lớn và độ chính xác cao, có thể nhận dạng đối tượng trong thời gian thực. Việc ứng dụng mô hình YOLO giúp giám sát, quản lý giao thông hiệu quả hơn, phát hiện và xử lý các vi phạm giao thông, đồng thời thúc đẩy nghiên cứu và ứng dụng công nghệ tiên tiến, góp phần giải quyết các vấn đề giao thông và phát triển bền vững xã hội.

1.3. MỤC TIÊU CỦA ĐỀ TÀI:

Phát triển một hệ thống nhận dạng biển báo giao thông tự động, có khả năng hoạt động hiệu quả trong điều kiện giao thông phức tạp của Việt Nam. Cụ thể, đề tài hướng đến việc xây dựng và huấn luyện mô hình YOLO để nhận dạng chính xác các loại biển báo giao thông khác nhau, từ đó cải thiện độ an toàn và hiệu quả trong việc tuân thủ luật giao thông. Đồng thời làm cơ sở cho việc tích hợp vào các hệ thống thực tế như xe thông minh, các thiết bị hỗ trợ người dùng khi tham gia giao thông. Cuối cùng, đề tài còn đóng góp vào việc thúc đẩy nghiên cứu và ứng dụng công nghệ AI trong lĩnh vực giao thông tại Việt Nam.

1.4. CẤU TRÚC ĐỀ TÀI:

Nội dung báo cáo được khai triển thành 5 chương:

Chương 1: Giới thiệu tổng quan về hướng nghiên cứu và mục tiêu nghiên cứu của đề tài.

Chương 2: Trình bày về tổng quan và cơ sở lý thuyết của Deep Learning, mô hình YOLO và bài toán nhận dạng đối tượng.

Chương 3: Triển khai đề tài: thu thập dữ liệu, gắn nhãn YOLO cho hình, tạo dataset, quá trình huấn luyện mô hình.

Chương 4: Đánh giá các kết quả về mô hình và đưa ra nhâ.

Chương 5: Đưa ra kết luận và hướng phát triển đề tài.

CHƯƠNG 2: TỔNG QUAN VỀ HỌC SÂU VÀ MÔ HÌNH MẠNG YOLO

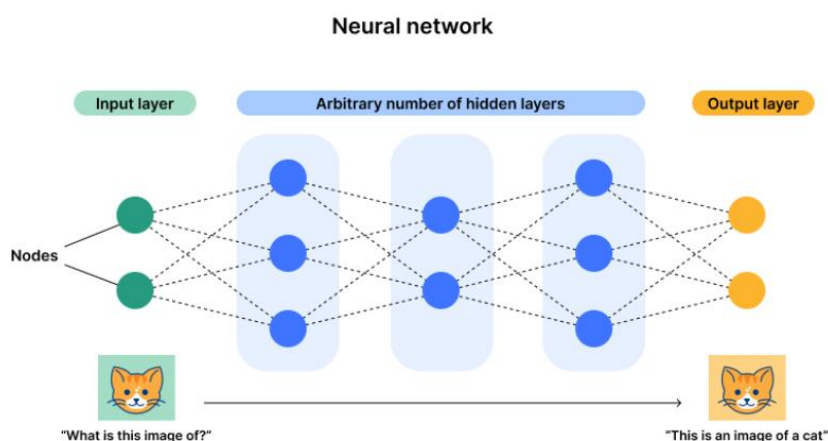
2.1. TỔNG QUAN VỀ HỌC SÂU:

Deep Learning, hay học sâu, là một lĩnh vực con của Machine Learning (học máy) trong trí tuệ nhân tạo (AI) tập trung vào việc sử dụng các mô hình mạng nơ-ron (Neural Networks) với nhiều lớp để học từ dữ liệu. Khác với các thuật toán học máy truyền thống, Deep Learning có khả năng tự động trích xuất các đặc trưng phức tạp từ dữ liệu thô, giúp giải quyết các bài toán khó khăn và phức tạp trong nhiều lĩnh vực.

2.1.1. Mạng Neural:

Mạng Neural là một loại kiến trúc tính toán dựa trên mô hình cách thức hoạt động của não con người – do đó có tên gọi là “Neural”. Mạng Neural được cấu tạo từ một tập hợp các đơn vị xử lý gọi là “nodes”. Các nodes này truyền dữ liệu cho nhau, giống như trong não bộ, các tế bào thần kinh truyền xung điện cho nhau.

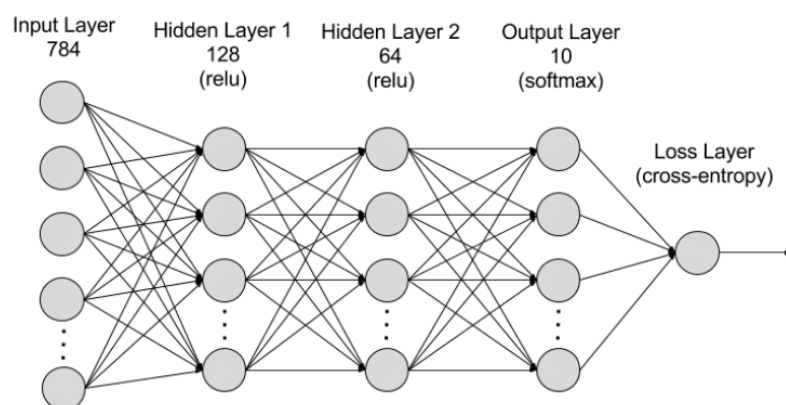
Mạng Neural được sử dụng trong Machine Learning, là một loại chương trình máy tính có khả năng học mà không cần các chỉ dẫn cụ thể. Mạng Neural được sử dụng trong Deep Learning – một loại Machine Learning tiên tiến có thể rút ra kết luận từ dữ liệu không được gán nhãn mà không cần sự can thiệp của con người. [1].



Hình 2-1: Minh họa cho mạng Neural Network

Kiến trúc mạng Neural đơn giản bao gồm các Neural nhân tạo liên kết theo 3 lớp:

- **Input Layer:** Thông tin từ bên ngoài đi vào mạng Neural qua input layer. Các node đầu vào xử lý dữ liệu, phân tích hoặc phân loại và sau đó chuyển dữ liệu sang lớp tiếp theo.
- **Hidden Layer:** Dữ liệu đi vào hidden layer đến từ input layer hoặc hidden layer khác. Mạng Neural có thể có một số lượng lớp hidden layer. Mỗi hidden layer phân tích dữ liệu đầu ra từ layer trước, xử lý dữ liệu đó sâu hơn và chuyển dữ liệu sang lớp tiếp theo.
- **Output Layer:** Output layer cho ra kết quả cuối cùng của tất cả các dữ liệu được xử lý bởi mạng Neural. Lớp này có thể có một hoặc nhiều node. Ví dụ: giả sử chúng ta gặp phải một vấn đề phân loại nhị phân (có/không), lớp đầu ra sẽ có một nút đầu ra, nút này sẽ cho kết quả 1 hoặc 0. Tuy nhiên, nếu chúng ta gặp phải vấn đề phân loại nhiều lớp, lớp đầu ra sẽ có thể bao gồm nhiều hơn một nút đầu ra.



Hình 2-2: Kiến trúc mạng Neural

2.1.2. Mạng Neural lan truyền ngược:

Mạng Neural lan truyền ngược (Backpropagation Neural Network) là một loại mạng Neural được huấn luyện thông qua thuật toán lan truyền ngược, một phương pháp quan trọng trong học máy và trí tuệ nhân tạo. Backpropagation là một thuật toán lặp lại, giúp

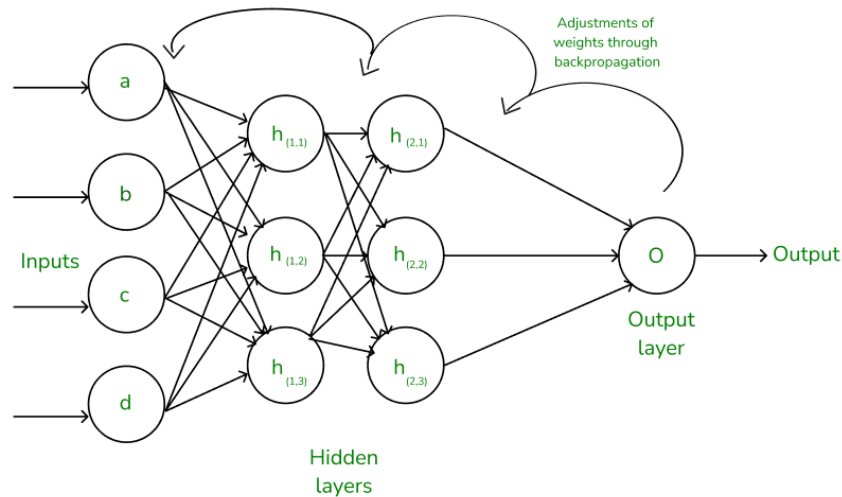
giảm thiểu chức năng chi phí bằng cách xác định trọng số và độ lệch nào cần được điều chỉnh. Đây là một trong những phương pháp phổ biến nhất để huấn luyện mạng Neural, cho phép mạng này học và điều chỉnh các trọng số để tối ưu hóa hiệu suất dự đoán [4].

Ưu điểm của mạng Neural lan truyền ngược:

Thuật toán Backpropagation là một công cụ quan trọng trong học máy và trí tuệ nhân tạo, mang lại nhiều ưu điểm đáng kể trong quá trình huấn luyện mạng Neural. Một trong những ưu điểm lớn nhất của Backpropagation là tính hiệu quả trong tính toán. Sử dụng quy tắc chuỗi, thuật toán này có khả năng tính toán gradient của hàm mất mát đối với các trọng số và bias một cách nhanh chóng và chính xác, giúp mạng Neural có thể được huấn luyện hiệu quả trên các mạng lớn và phức tạp.

Ngoài ra, Backpropagation còn mang lại khả năng học tự động từ dữ liệu. Bằng cách điều chỉnh các trọng số và bias dựa trên sai số dự đoán, mạng Neural có khả năng tự động học và cải thiện từ dữ liệu mà không cần sự can thiệp của con người. Điều này làm cho Backpropagation trở thành một công cụ mạnh mẽ cho việc giải quyết các bài toán phức tạp như nhận dạng hình ảnh, xử lý ngôn ngữ tự nhiên và dự đoán chuỗi thời gian. Thêm vào đó, Backpropagation còn mang lại tính linh hoạt và tiện lợi. Nó có thể được sử dụng với nhiều loại hàm mất mát và thuật toán tối ưu hóa khác nhau tùy thuộc vào bài toán cụ thể. Điều này giúp thuật toán có khả năng áp dụng cho nhiều loại bài toán khác nhau một cách dễ dàng và linh hoạt.

Cuối cùng, Backpropagation còn cho phép mạng Neural được huấn luyện một cách toàn diện qua nhiều vòng lặp. Với khả năng lặp lại qua các epochs, mạng Neural có thể cải thiện và học từ dữ liệu mới theo thời gian, tăng cường hiệu suất dự đoán và đảm bảo tính chính xác của mô hình. Tóm lại, Backpropagation là một công cụ mạnh mẽ và linh hoạt, mang lại nhiều lợi ích trong việc huấn luyện mạng neural và giải quyết các bài toán phức tạp trong học máy và trí tuệ nhân tạo.



Hình 2-3: Mạng Neural lan truyền ngược.

2.1.3. Một số hàm kích hoạt (Activation functions) phổ biến trong học sâu:

Hàm kích hoạt (activation function) mô phỏng tỷ lệ truyền xung qua axon của một neuron thần kinh. Trong một mạng nơ-ron nhân tạo, hàm kích hoạt đóng vai trò là thành phần phi tuyến tại output của các nơ-ron.

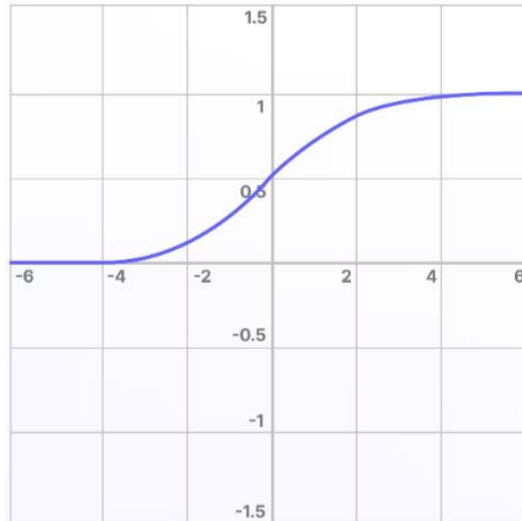
Hàm Sigmoid:

Cũng được biết đến là hàm kích hoạt logistic, nhận đầu vào và biến chúng thành đầu ra trong khoảng từ 0 đến 1. Vì lý do này, sigmoid được gọi là "hàm nén" và có khả năng khác biệt. Các giá trị đầu vào lớn hơn và tích cực sẽ cho ra các giá trị đầu ra gần 1.0, trong khi các giá trị đầu vào nhỏ hơn và tiêu cực sẽ cho ra các giá trị đầu ra gần 0.0.

Đây là hàm rất hữu ích đặc biệt trong các nhiệm vụ phân loại hoặc dự đoán xác suất, do đó có thể được triển khai trong quá trình huấn luyện các mạng thị giác máy tính và học sâu. Tuy nhiên, sự biến mất của gradient có thể gây ra vấn đề khi sử dụng trong các lớp ẩn, điều này có thể gây ra các vấn đề khi huấn luyện mô hình.

Công thức:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



Hình 2-4: Hàm Sigmoid

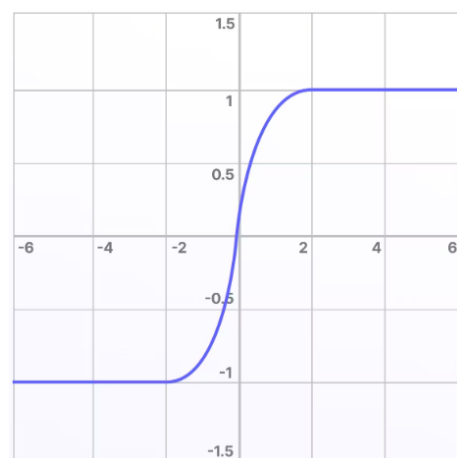
Hàm tanh:

Tanh (hay còn gọi là TanH), cũng được biết đến là hàm kích hoạt siêu bão hòa, tương tự như sigmoid/logistic, kể cả về hình dạng đường cong hình chữ S và có khả năng khác biệt. Tuy nhiên, ở đây, khoảng đầu ra của nó là từ -1 đến 1 (thay vì từ 0 đến 1). Đây là một độ dốc dốc hơn và cũng gặp phải thách thức của gradient biến mất như sigmoid/logistic.

Bởi vì các đầu ra của tanh có trung tâm ở giá trị zero-centric, các giá trị có thể dễ dàng được ánh xạ trên một thang đo từ rất tiêu cực, trung tính, đến rất tích cực.

Công thức:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



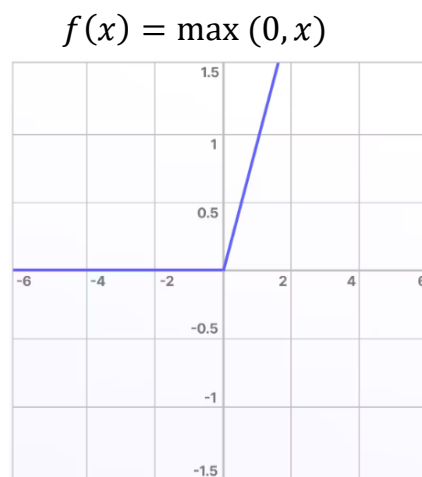
Hình 2-5: Hàm Tanh.

Hàm Rectified Linear Unit (ReLU):

Sau khi ReLU được nghiên cứu, hiệu suất huấn luyện đã tăng đáng kể. ReLU không có tính khả vi, nhưng điều này không phải là vấn đề vì đạo hàm có thể được tính toán cho ReLU.

ReLU không kích hoạt tất cả các nơ-ron theo trình tự cùng một lúc, làm cho nó hiệu quả hơn so với các hàm kích hoạt tanh hoặc sigmoid/logistic. Nhược điểm của điều này là một số trọng số và sai số cho các nơ-ron trong mạng có thể không được cập nhật hoặc kích hoạt.

Công thức:



Hình 2-6: Hàm ReLU.

Hàm Leaking ReLU:

Một giải pháp cho vấn đề "dying ReLU" là một biến thể được gọi là hàm kích hoạt Leaky ReLU. Với Leaky ReLU, thay vì giá trị là 0 khi $z < 0$, Leaky ReLU cho phép một độ dốc nhỏ không bằng không, được gọi là hằng số α (Thường là $\alpha = 0.01$).

Công thức:

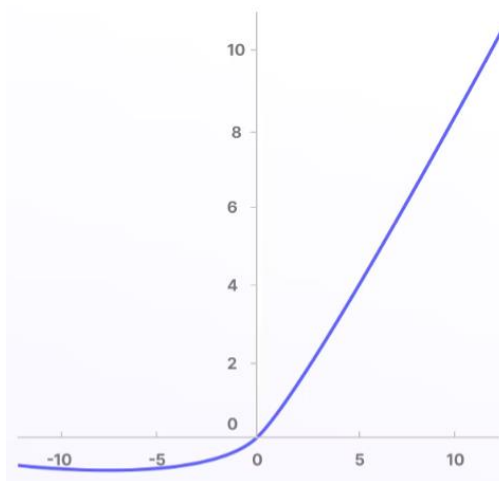
$$f(x) = 0.01x, x < 0; f(x) = x, x \geq 0$$



Hình 2-7: Hàm Leaking ReLU

Hàm Exponential Linear Unit (ELU):

ELU là một biến thể khác của hàm ReLU gốc, là một cách khác để vượt qua vấn đề "dying ReLU", và nó cũng không khả vi. ELUs sử dụng đường cong log cho các giá trị âm thay vì một đường thẳng, nó trở nên mượt mà dần cho đến khi đạt đến $-\alpha$. ^{[2], [3]}

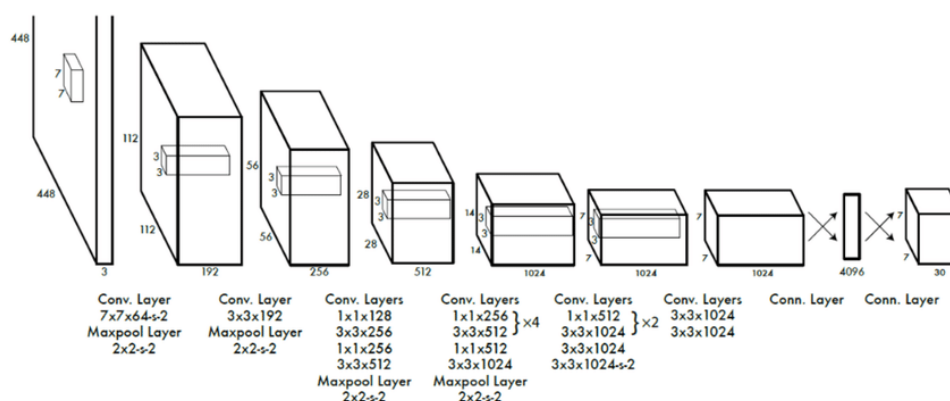


Hình 2-8: Hình ELU.

2.2. BÀI TOÁN VỀ NHẬN DẠNG ĐỐI TƯỢNG SỬ DỤNG YOLOV8:

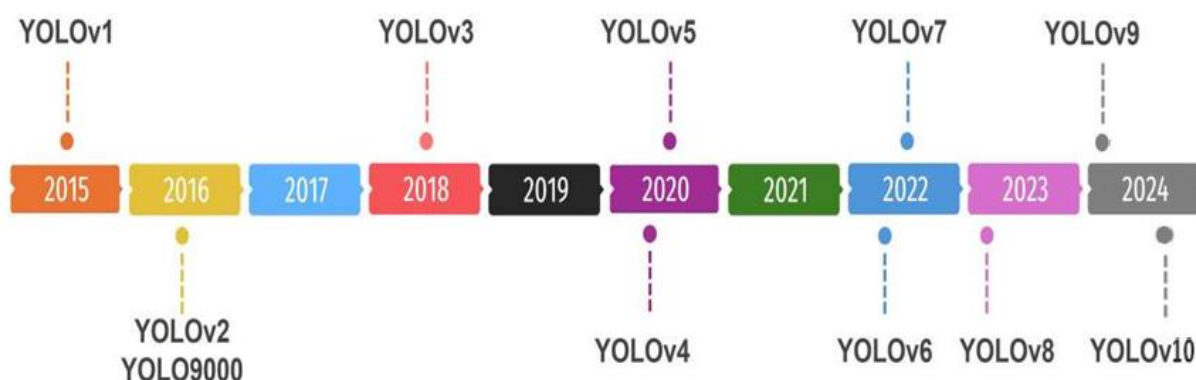
2.2.1. Giới thiệu YOLO:

YOLO là một mô hình mạng CNN cho việc phát hiện, nhận dạng, phân loại đối tượng. Yolo được tạo ra từ việc kết hợp giữa các convolutional layers và connected layers. Trong đó các convolutional layers sẽ trích xuất ra các feature của ảnh, còn full-connected layers sẽ dự đoán ra xác suất đó và tọa độ của đối tượng. [5]



Hình 2-9: Sơ đồ các lớp của mô hình YOLO cơ bản

Lịch sử phát triển của YOLO:



Hình 2-10: Lịch sử phát triển của YOLO

Ra mắt vào năm 2015, YOLO nhanh chóng trở nên phổ biến vì tốc độ và độ chính xác cao.

YOLOv2, được phát hành vào năm 2016, đã cải thiện mô hình ban đầu bằng cách kết hợp chuẩn hóa hàng loạt, hộp neo và cụm kích thước.

YOLOv3, ra mắt vào năm 2018, đã nâng cao hơn nữa hiệu suất của mô hình bằng cách sử dụng mạng đường trục hiệu quả hơn, nhiều neo và gộp kim tự tháp không gian.

YOLOv4 được phát hành vào năm 2020, giới thiệu những cải tiến như tăng cường dữ liệu Mosaic, đầu phát hiện không neo mới và chức năng mất mát mới.

YOLOv5 Cải thiện hơn nữa hiệu suất của mô hình và thêm các tính năng mới như tối ưu hóa siêu tham số, theo dõi thử nghiệm tích hợp và xuất tự động sang các định dạng xuất phổ biến.

YOLOv6 được Meituan mở nguồn mở vào năm 2022 và đang được sử dụng trong nhiều robot giao hàng tự động của công ty.

YOLOv7 đã thêm các tác vụ bổ sung như ước tính tư thế trên bộ dữ liệu điểm chính COCO.

YOLOv8 là phiên bản mới nhất của YOLO bằng cách Ultralytics. Là một mô hình tiên tiến, hiện đại (SOTA), YOLOv8 được xây dựng dựa trên sự thành công của các phiên bản trước, giới thiệu các tính năng và cải tiến mới để nâng cao hiệu suất, tính linh hoạt và hiệu quả. YOLOv8 hỗ trợ đầy đủ các tác vụ AI thị giác, bao gồm phát hiện, phân đoạn, ước tính tư thế, theo dõi và phân loại. Tính linh hoạt này cho phép người dùng tận dụng YOLOv8 với khả năng của nó trên các ứng dụng và lĩnh vực đa dạng.

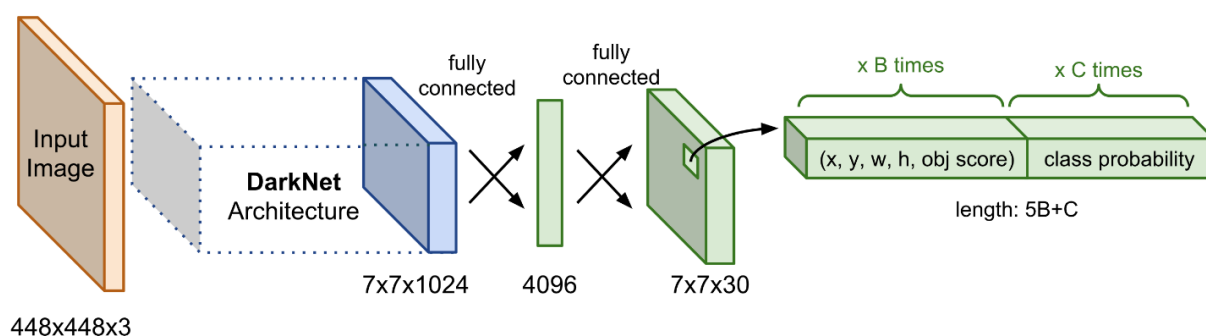
YOLOv9 Giới thiệu các phương pháp sáng tạo như Programmable Gradient Information (PGI) và Generalized Efficient Layer Aggregation Network (GELAN).

YOLOv10 ra mắt vào ngày 23/05/2024, YOLOv10 huấn luyện không cần Non-Maximum Suppression (NMS) trong quá trình suy luận, giúp giảm độ trễ và cải thiện hiệu suất. YOLOv10 được cải tiến kiến trúc mô hình, tăng cường độ chính xác và hiệu suất, tối ưu hoá cho việc sử dụng thời gian thực.

Kiến trúc mạng của mô hình YOLO:

Kiến trúc của mô hình YOLO bao gồm: base network là các mạng convolution và những Extra Layers được áp dụng để phát hiện vật thể trên feature map của base network.

Thuật toán YOLO lấy hình ảnh làm đầu vào và sau đó sử dụng mạng nơ-ron tích chập sâu đơn giản để phát hiện các đối tượng trong ảnh. Kiến trúc của mô hình CNN tạo thành xương sống của YOLO.

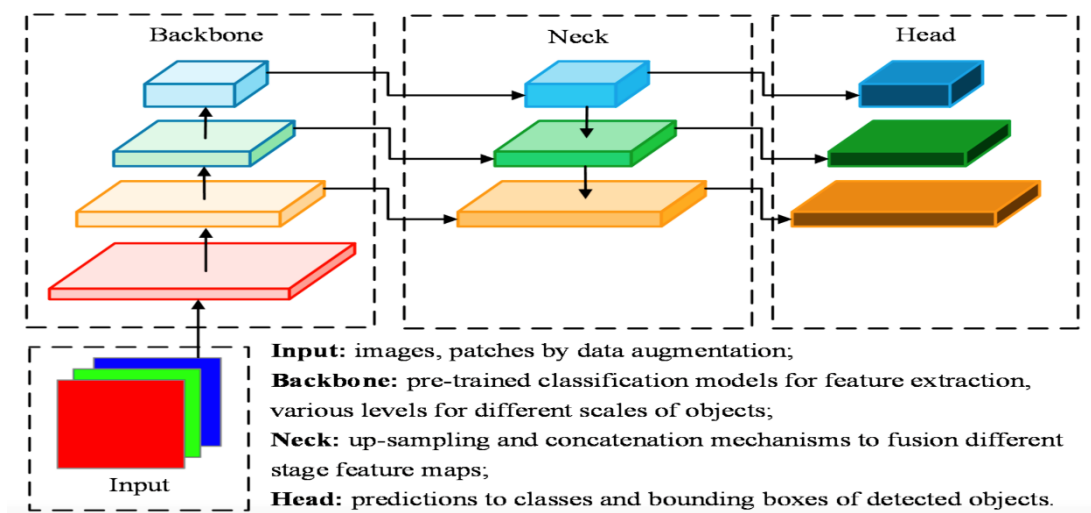


Hình 2-11: Sơ đồ kiến trúc mạng YOLO.

Kiến trúc mạng chung của các phiên bản mô hình YOLO

- **Input (Xử lý đầu vào):** Hình ảnh đầu vào được thay đổi kích thước thành kích thước cố định để đảm bảo tính nhất quán.
- **Phần Backbone (Xương sống):** YOLO sử dụng CNN làm xương sống để trích xuất các tính năng có ý nghĩa từ hình ảnh đầu vào. Các lựa chọn phổ biến cho đường trục bao gồm các biến thể của DarkNet hoặc ResNet.

- **Phần Head (Phần đầu):** Phần này chịu trách nhiệm dự đoán vị trí và lớp của các đối tượng trong hình ảnh. YOLO sử dụng các lớp YOLO để thực hiện nhiệm vụ này. Các lớp YOLO bao gồm các thao tác như dự đoán bounding box, dự đoán độ tin cậy cho mỗi lớp và thực hiện non-max suppression để loại bỏ các dự đoán trùng lặp.
- **Phần Neck (Cổ):** Ngoài 2 phần trên ra thì trong những mạng hiện nay thì nhiều tác giả bổ sung thêm phần Neck. Phần này được thêm vào giữa Head và Backbone để tăng cường sự phong phú và khả năng biểu diễn ngữ nghĩa của các đối tượng được trích xuất cho các đối tượng có hình dạng và kích thước khác nhau.
- **Output (Đầu ra):** Đầu ra cuối cùng của YOLO là một tập hợp các hộp giới hạn, điểm đối tượng liên quan của chúng và xác suất của lớp được dự đoán. Những kết quả này đại diện cho các đối tượng được phát hiện trong ảnh.

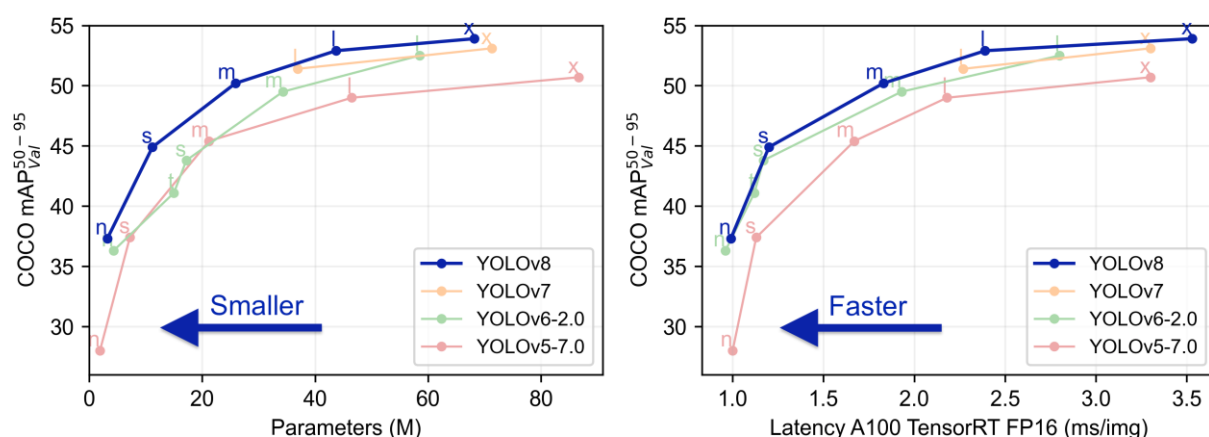


Hình 2-12: Kiến trúc mạng chung của mô hình mạng YOLO

2.2.2. Tổng quan về mô hình YOLOv8:

YOLOv8 là phiên bản mới nhất trong dòng máy dò tìm vật thể thời gian thực YOLO, mang lại hiệu suất vượt trội về độ chính xác và tốc độ. Dựa trên những tiến bộ của các phiên bản YOLO trước đó, YOLOv8 giới thiệu các tính năng và tối ưu hóa mới khiến nó

trở thành lựa chọn lý tưởng cho các tác vụ phát hiện đối tượng khác nhau trong nhiều ứng dụng.



Hình 2-13: So sánh về độ chính xác của YOLOv8 so với các phiên bản trước đó. [6]

Các tính năng chính:

Kiến trúc xương sống và cổ nâng cao: YOLOv8 sử dụng kiến trúc xương sống và cổ tiên tiến, giúp cải thiện hiệu suất trích xuất tính năng và phát hiện đối tượng.

Đầu Ultralytics phân chia không có neo: YOLOv8 sử dụng đầu Ultralytics phân chia không có neo, góp phần mang lại độ chính xác cao hơn và quy trình phát hiện hiệu quả hơn so với các phương pháp dựa trên neo.

Sự cân bằng giữa độ chính xác và tốc độ được tối ưu hóa: Với trọng tâm là duy trì sự cân bằng tối ưu giữa độ chính xác và tốc độ, YOLOv8 phù hợp cho các tác vụ phát hiện đối tượng theo thời gian thực trong các lĩnh vực ứng dụng đa dạng.

Nhiều mô hình được đào tạo trước: YOLOv8 cung cấp một loạt các mô hình được đào tạo trước để đáp ứng các nhiệm vụ và yêu cầu hiệu suất khác nhau, giúp bạn dễ dàng tìm thấy mô hình phù hợp cho trường hợp sử dụng cụ thể của mình.

Bảng 2-1: So sánh hiệu suất của các phiên bản YOLOv8 (nguồn Ultralytics)

Model	Size (pixels)	mAPval 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	Params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

2.2.3. Khái quát về hệ thống nhận dạng đối tượng:

Hệ thống nhận dạng đối tượng (Object Recognition System) là một hệ thống công nghệ sử dụng các phương pháp và kỹ thuật từ lĩnh vực học máy (machine learning), thị giác máy tính (computer vision) và xử lý ảnh (image processing) để xác định và phân loại các đối tượng trong hình ảnh hoặc video.

Các thành phần chính của hệ thống nhận dạng đối tượng:

Thu thập dữ liệu: Hệ thống nhận dữ liệu đầu vào dưới dạng hình ảnh hoặc video từ các thiết bị như máy ảnh, camera giám sát, hoặc các thiết bị ghi hình khác.

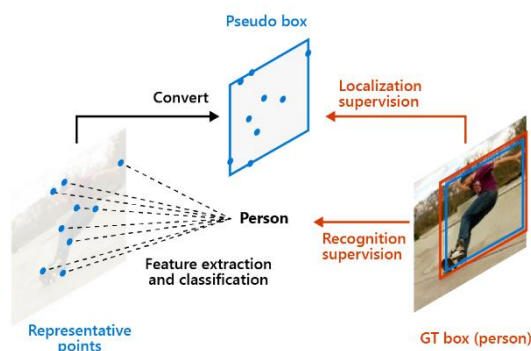
Tiền xử lý dữ liệu: Các hình ảnh hoặc video được xử lý để loại bỏ nhiễu, điều chỉnh độ sáng, độ tương phản và kích thước để phù hợp với các bước xử lý tiếp theo.

Trích xuất đặc trưng: Hệ thống trích xuất các đặc trưng (features) từ hình ảnh, chẳng hạn như cạnh, góc, màu sắc, hoặc các đặc trưng phức tạp hơn do các mạng nơ-ron tích chập (CNN) học được.

Phân loại đối tượng: Các đặc trưng được đưa vào mô hình phân loại (classifier) để xác định đối tượng trong hình ảnh. Các mô hình phổ biến bao gồm mạng nơ-ron sâu (DNN), mạng nơ-ron tích chập (CNN), và các phương pháp học máy truyền thống như SVM, cây quyết định, v.v.

Xác định và gán nhãn đối tượng: Hệ thống xác định các đối tượng và gán nhãn phù hợp cho mỗi đối tượng trong hình ảnh hoặc video.

Đánh giá và tinh chỉnh: Hệ thống được đánh giá dựa trên các tiêu chí như độ chính xác, độ nhạy, và độ đặc hiệu. Dữ liệu và mô hình có thể được tinh chỉnh và cải tiến liên tục dựa trên kết quả này.

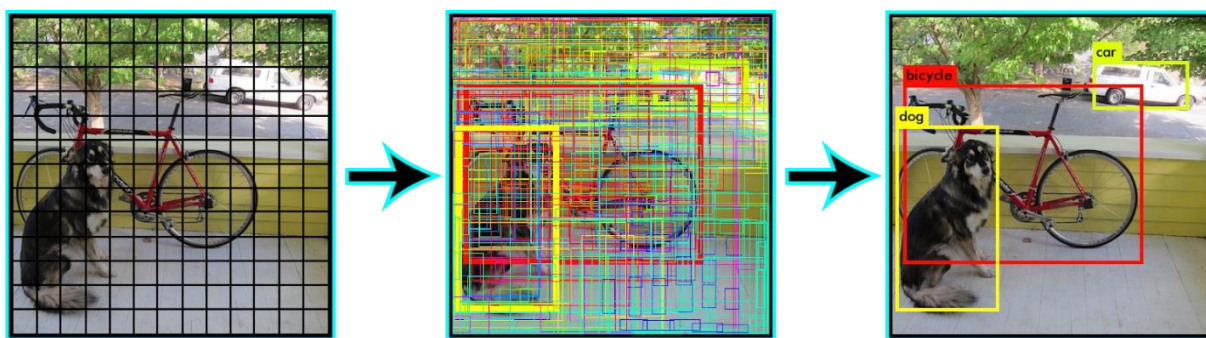


Hình 2-14: Minh họa cho hệ thống nhận dạng đối tượng.

2.2.4. Hệ thống nhận dạng đối tượng sử dụng YOLOv8:

Hệ thống nhận dạng đối tượng bằng mô hình YOLO (You Only Look Once) là một phương pháp tiên tiến và hiệu quả trong lĩnh vực thị giác máy tính, đặc biệt là trong việc phát hiện đối tượng. Khác với các phương pháp truyền thống cần nhiều bước để phát hiện đối tượng, YOLO thực hiện nhiệm vụ này trong một lần chạy duy nhất thông qua Convolutional Neural Network (CNN). Hệ thống chia ảnh đầu vào thành các lưới và dự

đoán bounding box cũng như các lớp xác suất cho mỗi lưới. YOLO được thiết kế để hoạt động nhanh chóng và hiệu quả, có thể xử lý hàng chục khung hình mỗi giây, làm cho nó phù hợp cho các ứng dụng yêu cầu phản hồi tức thì như giám sát an ninh, xe tự hành và robot. Thay vì chỉ xem các vùng nhỏ của ảnh như một số phương pháp khác, YOLO xem toàn bộ ảnh đầu vào trong một lần, giúp nó nhận dạng đối tượng một cách toàn diện và chính xác hơn. Sử dụng một mạng duy nhất để đưa ra dự đoán, YOLO giúp giảm thiểu lỗi phát hiện so với các phương pháp chia thành nhiều bước, đồng thời vẫn duy trì được độ chính xác cao nhờ vào thiết kế mạng hiệu quả và các cải tiến liên tục qua các phiên bản. YOLO được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau như giám sát an ninh, xe tự hành, y tế và robot, hỗ trợ các hệ thống này trong việc phát hiện, theo dõi và tương tác với môi trường xung quanh. Nhờ vào khả năng mạnh mẽ và linh hoạt, YOLO đã trở thành một công cụ quan trọng trong sự phát triển của thị giác máy tính và các ứng dụng trí tuệ nhân tạo khác.



Hình 2-15: Minh họa cho các hoạt động của YOLO trong nhận dạng đối tượng.

2.2.5. Một số khái niệm cơ bản liên quan tới bài toán nhận dạng đối tượng:

Bounding box (Hộp giới hạn):

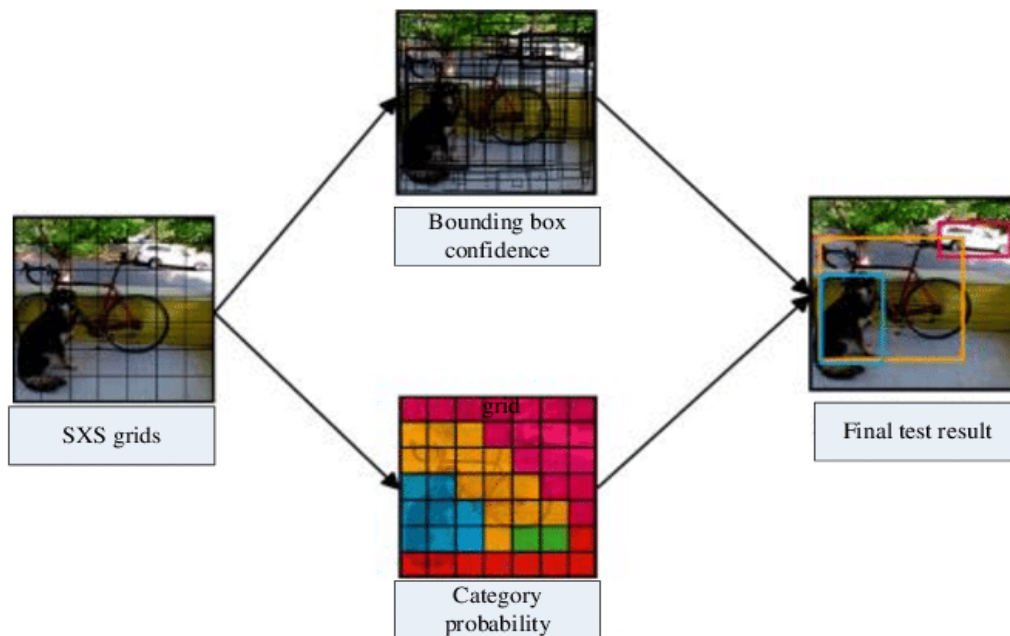
Bounding box (khối giới hạn hoặc vùng giới hạn) là một hình dạng hình học bao quanh một đối tượng hoặc một nhóm đối tượng trong hình ảnh kỹ thuật số.

Mục đích của hộp giới hạn là xác định vị trí và kích thước của đối tượng trong không gian 2D hoặc 3D, đồng thời cho phép thao tác và phân tích đối tượng dễ dàng.

Các hộp giới hạn trong hình ảnh 2D thường được thể hiện bằng hình chữ nhật, với các cạnh dài chạy song song với trục x và y của hình ảnh. Các hộp giới hạn trong hình ảnh 3D thường được biểu diễn bằng các hình song song (hình chữ nhật 3D), có các cạnh song song với các trục x, y và z của hình ảnh. [7]

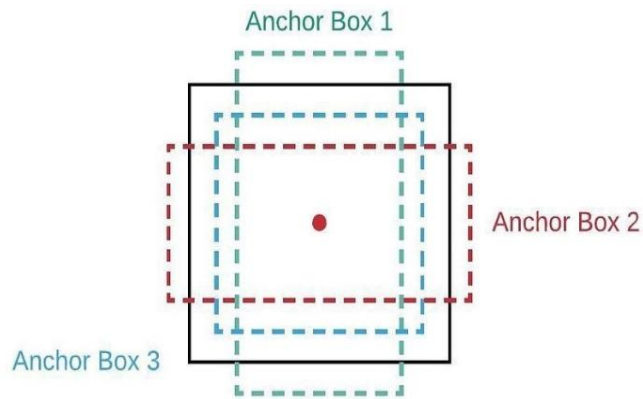
Anchor box (Hộp neo trước):

Anchor box là một tập hợp các hộp giới hạn được xác định trước có chiều cao và chiều rộng nhất định. Các hộp này được xác định để nắm bắt tỷ lệ và tỷ lệ khung hình của các lớp đối tượng cụ thể mà bạn muốn phát hiện và thường được chọn dựa trên kích thước đối tượng trong tập dữ liệu huấn luyện của bạn.



Hình 2-16: Minh họa cho hộp giới hạn.

Trong quá trình phát hiện, các hộp neo được xác định trước sẽ được xếp chồng lên nhau trên hình ảnh. Mạng dự đoán xác suất và các thuộc tính khác, chẳng hạn như nền, Intersection over Union (IoU) và độ lệch cho mỗi hộp neo xếp kè nhau. Các dự đoán được sử dụng để tinh chỉnh từng hộp neo riêng lẻ. Có thể xác định các hộp neo dành cho đối tượng khác nhau dựa theo kích thước. [7]

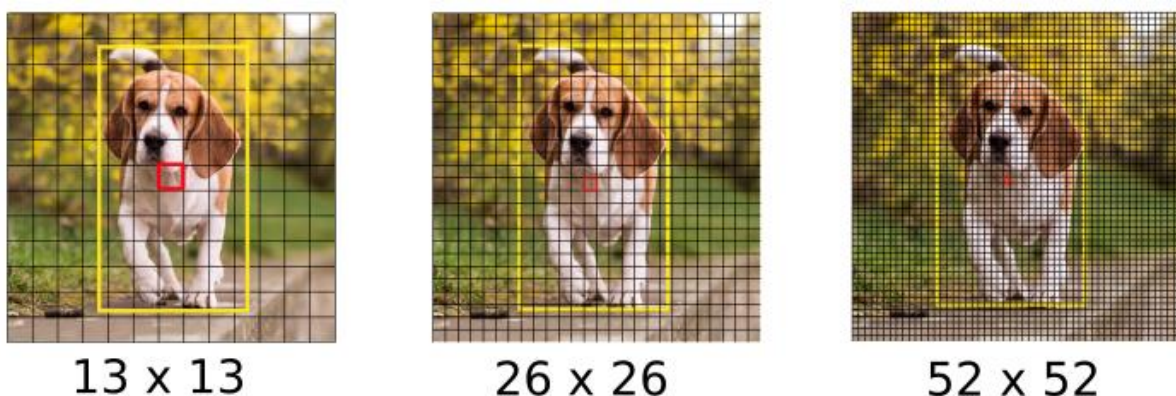


Hình 2-17: Minh họa hộp neo trước.

Bản đồ đặc trưng:

Bản đồ đặc trưng là kỹ thuật được sử dụng trong phân tích dữ liệu và máy học để chuyển đổi dữ liệu đầu vào từ không gian có chiều thấp hơn sang không gian có chiều cao hơn, nơi dữ liệu có thể được phân tích hoặc phân loại dễ dàng hơn.

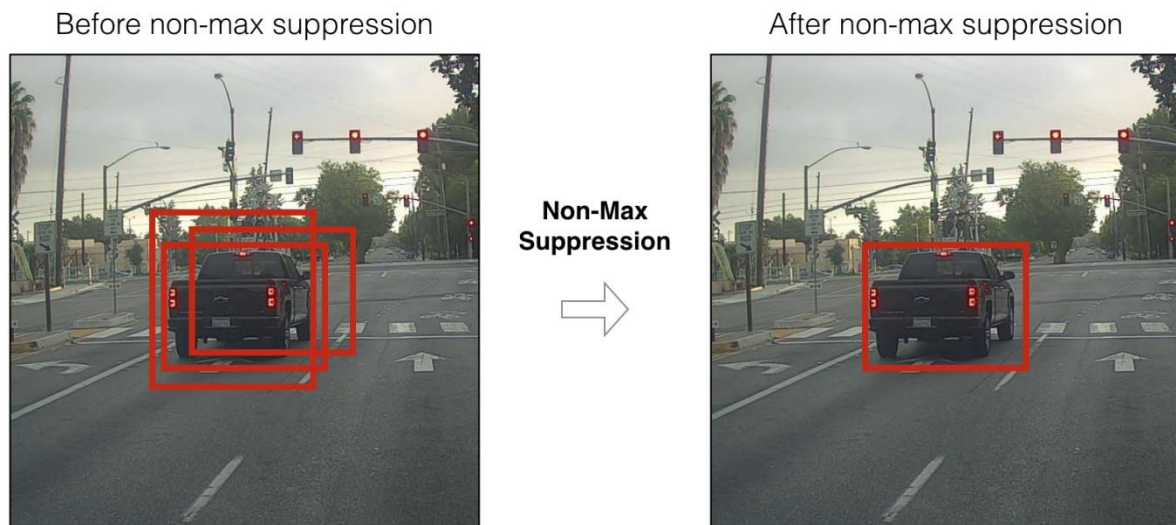
Ánh xạ đối tượng liên quan đến việc chọn hoặc thiết kế một tập hợp các hàm ánh xạ dữ liệu gốc sang một tập hợp các đối tượng mới để nắm bắt tốt hơn các mẫu cơ bản trong dữ liệu. Sau đó, không gian đặc trưng thu được có thể được sử dụng làm đầu vào cho thuật toán học máy hoặc kỹ thuật phân tích khác. Ánh xạ đối tượng có thể được sử dụng trong nhiều ứng dụng, xử lý ngôn ngữ tự nhiên đến thị giác máy tính và là một công cụ mạnh mẽ để chuyển đổi dữ liệu sang định dạng có thể phân tích dễ dàng hơn.^[8]



Hình 2-18: Các bản đồ đặc trưng cho hình

Thuật toán Non-Maximum Suppression

Thuật toán Non-Maximum Suppression (NMS) là một lớp thuật toán để chọn một thực thể (ví dụ: các hộp giới hạn) trong số nhiều thực thể chồng chéo. Chúng ta có thể lựa chọn các tiêu chí lựa chọn để đi đến kết quả mong muốn. Các tiêu chí phổ biến nhất là một số dạng số xác suất và một số dạng đo lường chồng chéo. Nó được sinh ra để loại bỏ đi các bounding box dư thừa của cùng một đối tượng tượng trong ảnh.



Hình 2-19: Từ 3 bounding box ban đầu cùng bao quanh chiếc xe đã giảm xuống còn một bounding box cuối cùng

Các bước để thực hiện thuật toán NMS:

Bước 1: Đầu tiên chúng ta sẽ tìm cách giảm bớt số lượng các bounding box bằng cách lọc bỏ toàn bộ những bounding box có xác suất chứa vật thể nhỏ hơn một ngưỡng threshold nào đó, thường là 0.5.

Bước 2: Đối với các bounding box giao nhau, NMS sẽ lựa chọn ra một bounding box có xác suất chứa vật thể là lớn nhất. Sau đó tính toán chỉ số giao thoa IoU với các bounding box còn lại. Nếu chỉ số này lớn hơn ngưỡng threshold thì điều đó chứng tỏ 2 bounding boxes đang overlap nhau rất cao. Ta sẽ xóa các bounding có xác suất thấp hơn và giữ

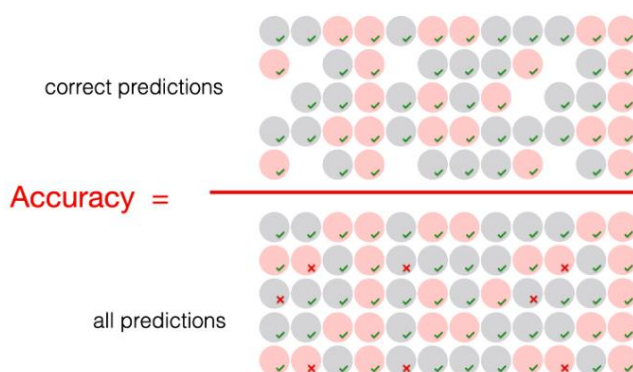
lại bounding box có xác suất cao nhất. Cuối cùng, ta thu được một bounding box duy nhất cho một vật thể.

2.2.6. Một số tham số đánh giá hiệu suất mô hình:

Accuracy (Độ chính xác):

Độ chính xác là tỷ lệ phần trăm của các dự đoán đúng trên tổng số dự đoán. Tuy nhiên, trong bối cảnh nhận dạng đối tượng, độ chính xác không phải lúc nào cũng là thước đo tốt nhất, đặc biệt khi đối mặt với các vấn đề như dữ liệu không cân bằng hoặc yêu cầu độ chính xác cao về vị trí của các đối tượng. ^[9]

$$Accuracy = \frac{Correct\ predictions}{All\ predictions}$$

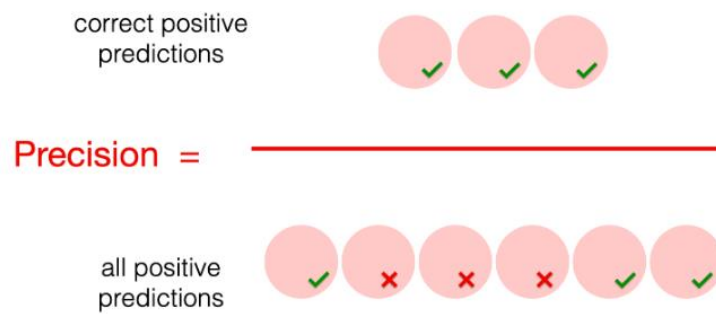


Hình 2-20: Hình minh họa cho độ chính xác.

Precision (Độ chuẩn xác):

Precision cho biết tỷ lệ phần trăm của các dự đoán dương đúng trong tổng số dự đoán dương. ^[10]

$$Precision = \frac{Correct\ positive\ predictions}{All\ positive\ predictions}$$



Hình 2-21: Hình minh họa cho độ chuẩn xác

Recall (Độ nhạy):

Recall cho biết tỷ lệ phần trăm của các đối tượng dương thật sự được mô hình nhận diện đúng. [8]

$$Recall = \frac{True\ positives}{True\ positives + False\ Negatives}$$

F1 Score:

F1 Score là trung bình điều hòa của precision và recall, cung cấp một thước đo cân bằng khi có sự đánh đổi giữa precision và recall.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Box loss còn được gọi là loss function cho bounding boxes. Đặc biệt, trong các bài toán như phát hiện đối tượng (object detection), box loss thường liên quan đến việc tính toán sự khác biệt giữa các bounding box dự đoán và các bounding box thực tế.

Class loss: Tổn thất liên quan đến việc phân loại đúng các lớp của vật thể trong các hộp bao quanh, sử dụng hàm tổn thất cross-entropy.

mAP50 và mAP50-95:

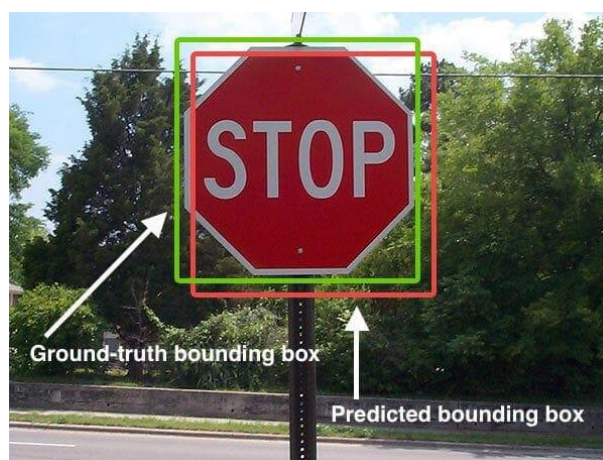
Bài toán Object Detection của chúng ta có nhiều class, mỗi class ta sẽ tiến hành đo AP, sau đó lấy trung bình của tất cả các giá trị AP của các class ta được chỉ số mAP của mô hình.

mAP50: Độ chính xác trung bình được tính ở ngưỡng giao nhau trên giao nhau (IoU) là 0.50. Đó là thước đo độ chính xác của mô hình chỉ xem xét các phát hiện dễ dàng.

mAP50-95: Trung bình của độ chính xác trung bình được tính ở các ngưỡng IoU khác nhau, dao động từ 0.50 đến 0.95. Nó cung cấp một cái nhìn toàn diện về hiệu suất của mô hình qua các mức độ khó phát hiện khác nhau. ^[11]

Intersection over Union (IOU):

Là hàm đánh giá độ chính xác của object detector trên tập dữ liệu cụ thể và được sử dụng để đánh giá hiệu suất phát hiện đối tượng bằng cách so sánh giữa ground truth bounding box và predicted bounding box.



Hình 2-22: Minh họa cho xác định các bounding box.

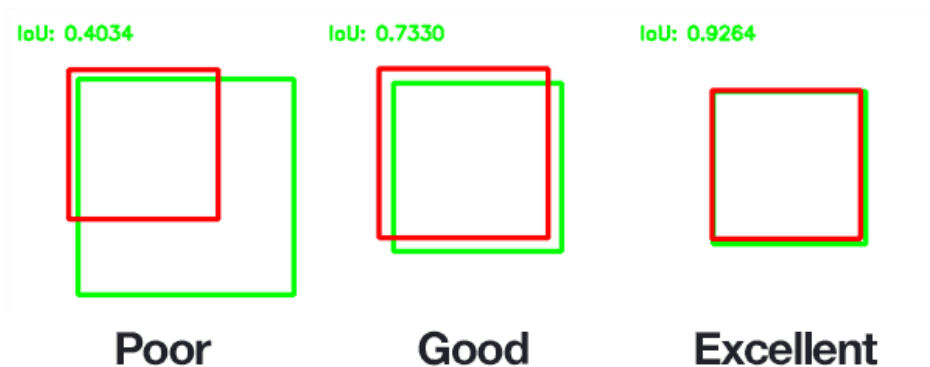
Vùng màu xanh là ground-truth bounding box (có xác định bằng cách đánh nhãn bằng tay), vùng màu đỏ là predicted bounding box là vùng dự đoán được từ mô hình.

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

Hình 2-23: Công thức tính IOU

Trong đó Area of Overlap là diện tích phần giao nhau giữa predicted bounding box với growth-truth bouding box, còn Area of Union là diện tích phần hợp giữa predicted bounding box với growth-truth bounding box.

Nếu $\text{IOU} > 0.5$ thì prediction được đánh giá là tốt.



Hình 2-24: Các mức độ đánh giá sử dụng IOU.

2.3. GOOGLE COLAB, KAGGLE, ROBOFLOW VÀ ULTRALYTICS:

2.3.1. Google Colab:

Google Colab, hay Google Colaboratory, là một nền tảng đám mây tiên tiến do Google phát triển, nhằm hỗ trợ người dùng viết và chạy code Python một cách dễ dàng và hiệu quả. Google Colab sử dụng giao diện Jupyter Notebook, cho phép người dùng thực hiện các tác vụ phân tích dữ liệu, xây dựng mô hình Machine Learning, và nghiên cứu khoa

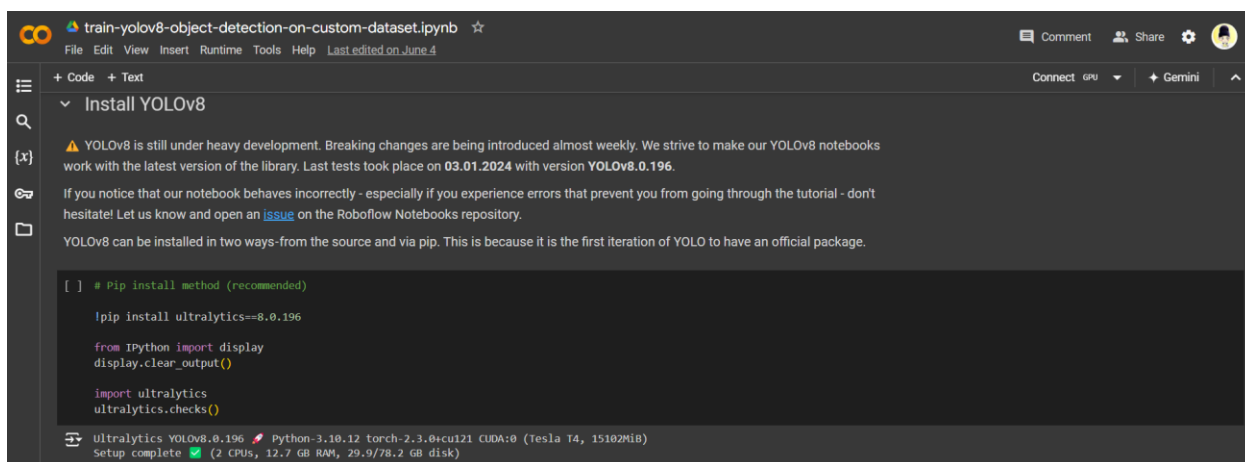
học một cách trực quan và tiện lợi. Một trong những điểm nổi bật của Google Colab là việc cung cấp miễn phí tài nguyên tính toán mạnh mẽ, bao gồm cả GPU và TPU, giúp tăng tốc đáng kể quá trình xử lý các bài toán phức tạp mà không yêu cầu cấu hình phần cứng cá nhân cao cấp.



Hình 2-25: Logo của Google Colab

Ngoài việc cung cấp khả năng tính toán mạnh mẽ, Google Colab còn tích hợp chặt chẽ với Google Drive, tạo điều kiện thuận lợi cho việc lưu trữ và chia sẻ các notebook. Người dùng có thể dễ dàng mở, chỉnh sửa và lưu các tệp notebook trực tiếp từ Google Drive, cũng như chia sẻ chúng với đồng nghiệp hoặc cộng đồng một cách đơn giản. Các thư viện Python phổ biến như NumPy, pandas, TensorFlow, Keras, và PyTorch được cài đặt sẵn, giúp người dùng bắt đầu các dự án khoa học dữ liệu và Machine Learning mà không cần phải trải qua quá trình cài đặt phức tạp.

Google Colab không chỉ phục vụ cho các cá nhân mà còn hỗ trợ mạnh mẽ cho làm việc nhóm. Tính năng collaborative coding cho phép nhiều người cùng làm việc trên một notebook, giống như cách làm việc trên Google Docs. Điều này rất hữu ích trong môi trường giáo dục, nghiên cứu và phát triển, nơi mà sự hợp tác và chia sẻ thông tin là cực kỳ quan trọng. Với tất cả những tính năng và tiện ích này, Google Colab đã trở thành một công cụ không thể thiếu trong cộng đồng khoa học dữ liệu và AI, giúp tối ưu hóa quá trình phát triển và triển khai các mô hình Machine Learning. ^[12]



Hình 2-26: Giao diện làm việc của Google Colab.

2.3.2. Kaggle:

Kaggle là một nền tảng trực tuyến dành riêng cho cộng đồng Machine Learning (ML) và Khoa học dữ liệu, nơi người dùng có thể chia sẻ và tìm kiếm các bộ dữ liệu phong phú, học hỏi và xây dựng các mô hình, đồng thời tương tác với các nhà khoa học và kỹ sư ML từ khắp nơi trên thế giới. Nền tảng này không chỉ cung cấp kho tàng dữ liệu đa dạng mà còn tạo điều kiện cho người dùng tham gia các cuộc thi khoa học dữ liệu, nơi họ có thể thử sức giải quyết các vấn đề thực tiễn và có cơ hội giành những giải thưởng giá trị. Thông qua các cuộc thi này, người dùng không chỉ nâng cao kỹ năng chuyên môn mà còn có thể xây dựng mạng lưới kết nối với các chuyên gia trong ngành.

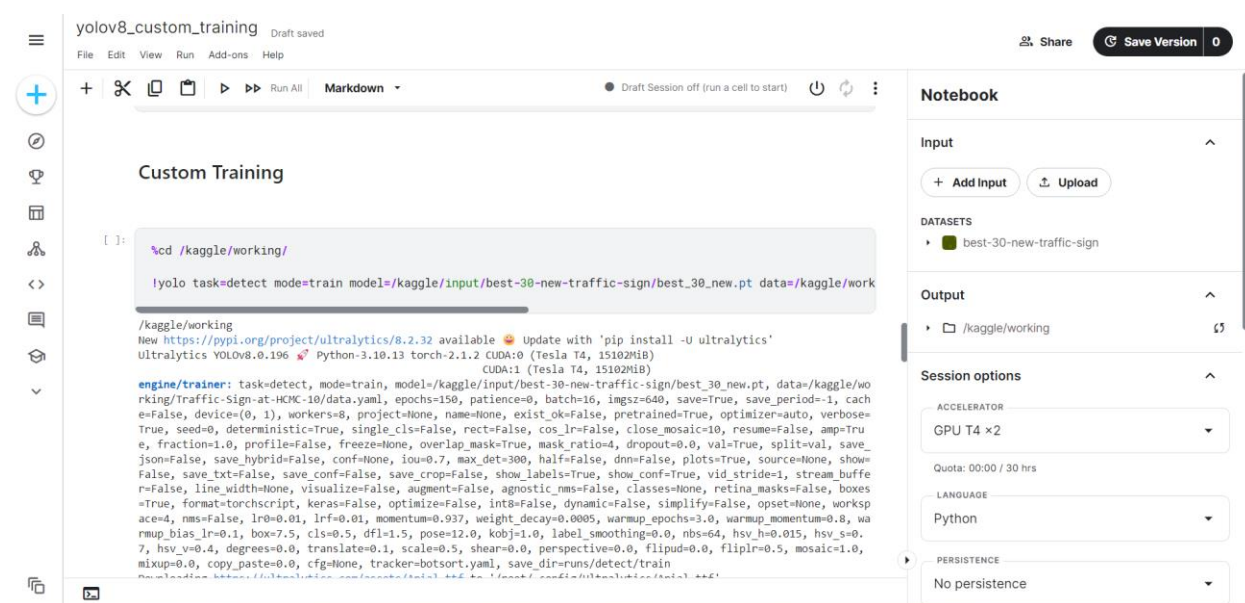


Hình 2-27: Logo của Kaggle

Kaggle còn hỗ trợ mạnh mẽ cho quá trình phát triển mô hình bằng cách cung cấp Graphic Processing Unit (GPU) và gần đây là Tensor Processing Unit (TPU), giúp tăng tốc độ tính toán trong quá trình huấn luyện và suy luận. Việc sử dụng GPU và TPU giúp rút ngắn đáng kể thời gian xử lý, cho phép người dùng thử nghiệm nhiều mô hình và tham số hơn trong khoảng thời gian ngắn. Bên cạnh đó, Kaggle còn cung cấp các công cụ lập trình

trực tuyến như Kaggle Kernels (nay gọi là Kaggle Notebooks), nơi người dùng có thể viết và chạy mã trực tiếp trên trình duyệt mà không cần cài đặt phần mềm bổ sung. Điều này tạo điều kiện thuận lợi cho việc phân tích dữ liệu, trực quan hóa và xây dựng mô hình học máy.

Ngoài ra, Kaggle còn là nơi người dùng có thể tìm thấy nhiều tài liệu học tập phong phú, từ các bài viết blog, bài giảng video, đến các khóa học trực tuyến, giúp họ liên tục cập nhật và nâng cao kiến thức. Các diễn đàn thảo luận sôi nổi trên Kaggle cũng là nơi lý tưởng để trao đổi kinh nghiệm, tìm kiếm giải pháp cho các vấn đề khó khăn và nhận được sự hỗ trợ từ cộng đồng. Với tất cả những tính năng và tài nguyên phong phú này, Kaggle thực sự là một công cụ quan trọng và phổ biến, giúp người dùng từ khắp nơi trên thế giới phát triển kỹ năng, chia sẻ kiến thức và tiến bộ trong lĩnh vực Machine Learning và Khoa học dữ liệu.



Hình 2-28: Giao diện làm việc của Kaggle

2.3.3. Roboflow:

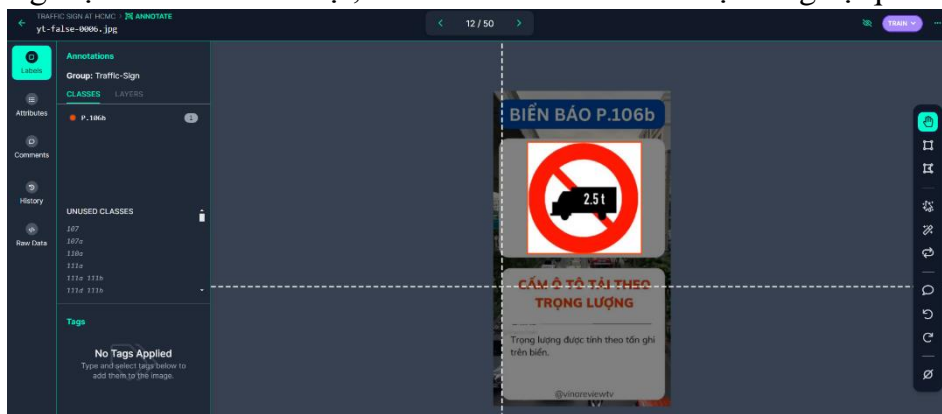
Roboflow là một nền tảng đám mây tiên tiến được thiết kế để hỗ trợ quá trình phát triển và triển khai các mô hình thị giác máy tính (computer vision). Với Roboflow, người dùng có thể dễ dàng thu thập, tổ chức, và gắn nhãn dữ liệu hình ảnh, một bước quan trọng

trong việc xây dựng các mô hình học sâu (deep learning) chất lượng cao. Nền tảng này hỗ trợ nhiều định dạng dữ liệu và cung cấp các công cụ mạnh mẽ để tiền xử lý và tăng cường dữ liệu (data augmentation), giúp cải thiện độ chính xác và hiệu suất của mô hình. Roboflow cũng tích hợp chặt chẽ với các framework machine learning phổ biến như TensorFlow, PyTorch, và Keras, giúp người dùng dễ dàng xuất dữ liệu và mô hình để huấn luyện và triển khai.



Hình 2-29: Logo của Roboflow.

Không chỉ dừng lại ở việc chuẩn bị dữ liệu, Roboflow còn cung cấp các giải pháp tối ưu hóa và triển khai mô hình, giúp các nhà phát triển nhanh chóng đưa các ứng dụng thị giác máy tính vào thực tế. Nền tảng này hỗ trợ việc triển khai trên nhiều môi trường khác nhau, từ các thiết bị IoT cho đến các hệ thống máy chủ lớn, đảm bảo tính linh hoạt và khả năng mở rộng. Bên cạnh đó, Roboflow cung cấp một cộng đồng người dùng và tài liệu phong phú, giúp người dùng mới dễ dàng bắt đầu và tận dụng tối đa các tính năng của nền tảng. Với những tính năng mạnh mẽ và linh hoạt, Roboflow đã trở thành một công cụ quan trọng trong lĩnh vực thị giác máy tính, giúp đẩy nhanh quá trình nghiên cứu và ứng dụng công nghệ này vào đời sống. [13]



Hình 2-30: Giao diện làm việc của Roboflow.

2.3.4. Ultralytics

Ultralytics là một công ty công nghệ tiên phong trong lĩnh vực trí tuệ nhân tạo và Machine Learning, đặc biệt nổi bật trong thị giác máy tính. Công ty này nổi tiếng với việc phát triển và duy trì các phiên bản của mô hình YOLO, YOLOv8 đại diện cho bước tiến vượt bậc trong phát hiện đối tượng, với cải tiến đáng kể về hiệu suất, tốc độ xử lý và độ chính xác. Mô hình này được thiết kế với tính thân thiện cao, dễ sử dụng và tinh chỉnh, phù hợp cho cả người mới bắt đầu và các chuyên gia. Bên cạnh đó, Ultralytics cung cấp Ultralytics HUB, một nền tảng toàn diện để quản lý dự án AI, bao gồm quản lý dữ liệu, huấn luyện mô hình, theo dõi hiệu suất và triển khai mô hình vào các ứng dụng thực tế, giúp người dùng tối ưu hóa quy trình làm việc của mình.

Ngoài việc phát triển các mô hình AI tiên tiến như YOLO, Ultralytics còn tập trung mạnh mẽ vào việc hỗ trợ và phát triển cộng đồng người dùng. Công ty cung cấp một loạt các tài liệu học tập chi tiết, từ hướng dẫn sử dụng, mã nguồn mở đến các bài viết blog và video hướng dẫn, giúp người dùng dễ dàng tiếp cận và nắm bắt công nghệ. Ultralytics duy trì một cộng đồng sôi nổi, nơi người dùng có thể trao đổi kinh nghiệm và tìm kiếm sự hỗ trợ. Các sản phẩm của Ultralytics, đặc biệt là YOLOv8, được ứng dụng rộng rãi trong nhiều lĩnh vực như an ninh, y tế và tự động hóa, với các ứng dụng cụ thể như giám sát an ninh, phân tích hình ảnh y tế và phát triển hệ thống tự động nhận diện và phân loại đối tượng. Với cam kết mạnh mẽ về mã nguồn mở, Ultralytics cho phép cộng đồng truy cập, sử dụng và đóng góp vào sự phát triển của các mô hình này, thúc đẩy tiến bộ trong nghiên cứu và ứng dụng thực tế của AI.

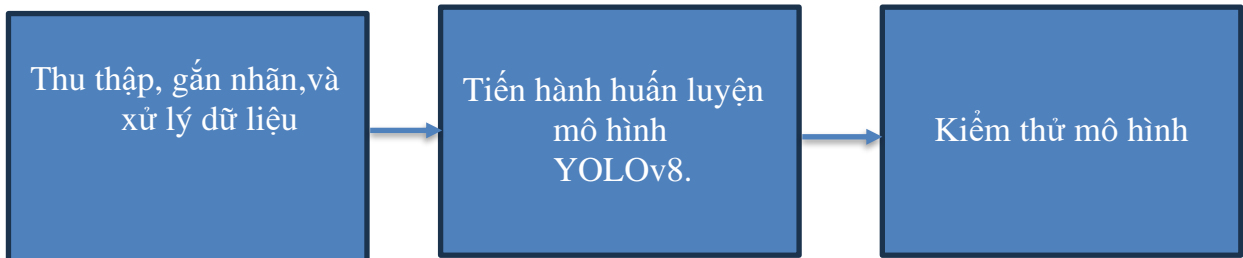


Hình 2-31: Logo của Ultralytics

CHƯƠNG 3: TRIỂN KHAI ĐỀ TÀI

3.1. SƠ ĐỒ KHỞI QUÁ TRÌNH TRIỂN KHAI:

Hệ thống nhận dạng các biển báo giao thông ở Việt Nam được tiến hành như sau:



3.1.1. Quá trình thu thập, gán nhãn và xử lý dữ liệu:

Quá trình thu thập, gán nhãn và xử lý dữ liệu là bước quan trọng trong việc xây dựng và triển khai một hệ thống nhận dạng đối tượng.

Nhóm chúng em tự tiến hành xây dựng tập dữ liệu. Nhóm đã tiến hành đi chụp các biển báo tại các con đường trên địa bàn thành phố Hồ Chí Minh. Ngoài ra, chúng em còn tải về một số video từ nguồn Youtube về các biển báo để tăng cường lượng dữ liệu thuận lợi cho quá trình huấn luyện.

Một số ảnh chụp minh họa:



Hình 3-1: Một số hình ảnh chụp các biển báo từ nhóm.

Nhóm chúng em đã thu thập gần hơn 10000 tấm hình với 59 loại biển báo thông dụng khác nhau.

Bảng 3-1: Tổng hợp 59 loại biển báo thông dụng và mã số của các loại biển báo.

Tên biển	Mã hiệu
Bến xe buýt	434a
Bệnh viện	425
Chỉ cho phép đi thẳng	301a
Chỉ hướng đường	1.414
Chỗ ngoặt bên phải	201b
Chỗ ngoặt bên trái	201a
Chỗ ngoặt nguy hiểm liên tiếp bên phải	202b
Chỗ ngoặt nguy hiểm liên tiếp bên trái	202a
Chỗ quay xe	409
Các xe chỉ được rẽ phải	301b
Các xe chỉ được rẽ trái	301c
Công trường	227
Cấm bóp còi	128
Cấm dừng xe và đỗ xe	130
Cấm người đi bộ	112
Cấm quay xe	124a1
Cấm rẽ phải	123b
Cấm rẽ trái	123a
Cấm rẽ trái quay đầu	124c
Cấm rẽ trái và rẽ phải	137
Cấm vượt	125
Cấm xe 3 bánh và xe máy	111a 111b
Cấm xe khách	107a
Cấm xe khách và xe tải	107
Cấm xe máy	111a
Cấm xe tải	P.106a
Cấm xe tải có khối lượng	P.106b
Cấm xe đạp	110a
Cấm ô tô	P.103a
Cấm ô tô quay đầu	124b1
Cấm ô tô rẽ phải	P.103c
Cấm ô tô rẽ trái	P.103b
Cấm ô tô rẽ trái và quay đầu	124e
Cấm đi ngược chiều	P.102
Cấm đỗ xe	131a

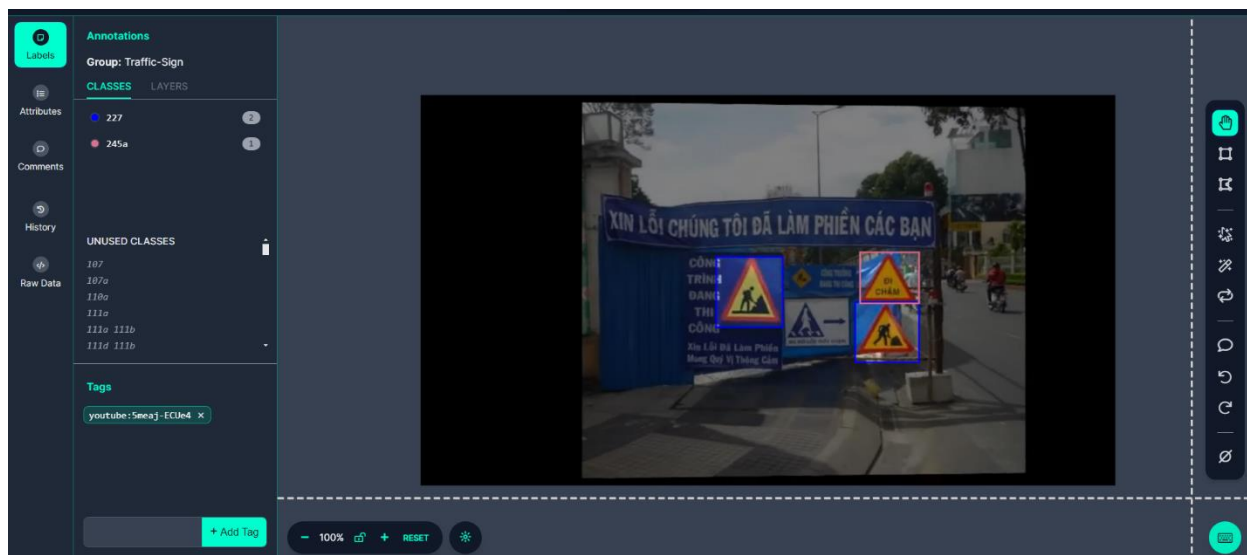
Cấm đỗ xe áp dụng với ngày chẵn	131c
Cấm đỗ xe áp dụng với ngày lẻ	131b
Cầu cho người đi bộ	1.424
Giao nhau đường không ưu tiên	207
Giao nhau đường ưu tiên	208
Hướng đi mỗi làn xe theo vạch kẻ đường	R.411
Hạn chế chiều cao xe	117
Người đi bộ cắt ngang	224
Người đi bộ sang đường	423b
Nơi giao nhau chạy theo vòng xuyên	303
Nơi đỗ xe	408
Phân làn	415a
Trẻ em	225
Tốc độ giới hạn 40 km/h	127-40
Tốc độ giới hạn 50 km/h	127-50
Tốc độ giới hạn 60 km/h	127-60
Tốc độ giới hạn 80 km/h	127-80
Đi chậm	245a
Đường 1 chiều	407a
Đường có gờ giảm tốc	221b
Đường cấm	P.101
Đường giao nhau	205
Đường đi vòng chướng ngại vật bên phải	302a
Đường đi vòng chướng ngại vật bên trái	302b

Sau khi thu thập dữ liệu xong, nhóm chúng em tiến hành dán nhãn YOLO cho các hình. Nhãn YOLO thường sẽ được lưu trữ dưới dạng file có đuôi *.txt. Tập phải được định dạng với một hàng cho mỗi đối tượng trong **class x_center y_center width height**. Tọa độ hộp phải ở trong XYWH chuẩn hóa định dạng (từ 0 đến 1). Nếu hộp đang ở dạng pixel, chúng ta nên chia **x_center** và **width** theo chiều rộng hình ảnh và **y_center** và **height** theo chiều cao hình ảnh. Số lớp phải được lập chỉ mục bằng 0 (bắt đầu bằng 0).



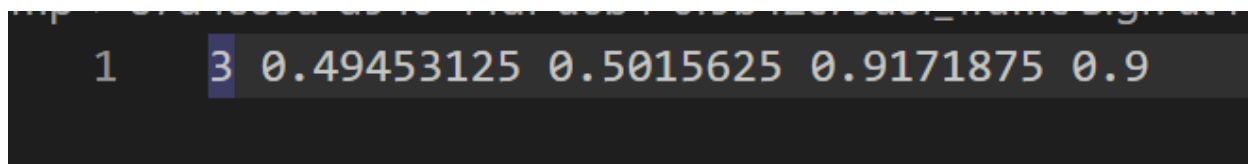
Hình 3-2: Minh họa cho các tham số nhãn YOLO

Việc dẫn nhãn, nhóm chúng em sử dụng công cụ Roboflow, một công cụ rất tiện ích trong Thị giác máy tính. Nó cho phép gắn nhãn và xuất ra được file *.txt theo nhãn YOLO. Và nó là một công cụ để lưu trữ tập dữ liệu và có thể liên kết tập dữ liệu thông qua một vài đoạn code. Tập dữ liệu sẽ được tải về để huấn luyện mô hình.



Hình 3-3: Quá trình dán nhãn cho các biển báo.

Cấu trúc file nhãn YOLO: **class x_center y_center width height**



Hình 3-4: Cấu trúc file nhãn của YOLO

Đoạn mã tải tập dữ liệu:

```
!pip install roboflow
```

```
from roboflow import Roboflow
```

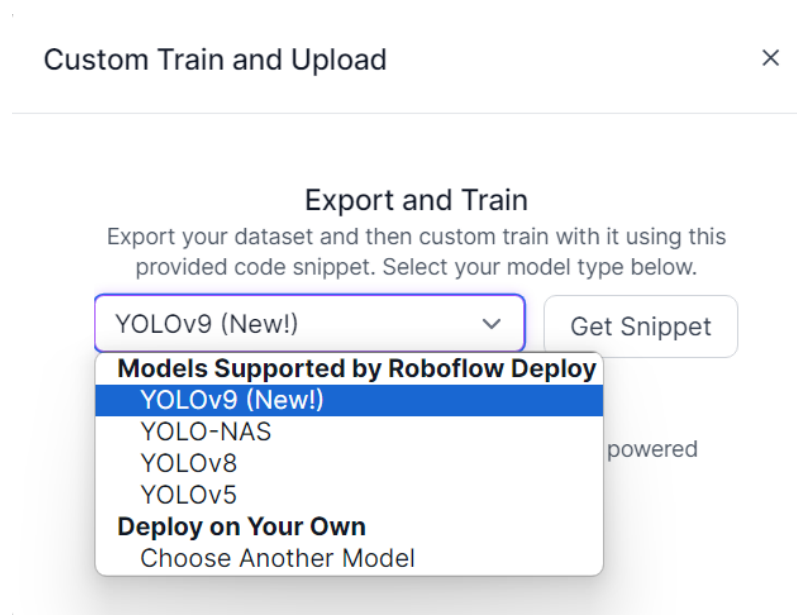
```
rf = Roboflow(api_key="3bnGDKd7LZf6w8OLQzn1")
```

```
project = rf.workspace("bao-dphco").project("traffic-sign-at-hcmc")

version = project.version(21)

dataset = version.download("yolov8")
```

Chúng ta có thể tạo ra rất nhiều loại file nhãn cho các loại mô hình YOLO khác nhau:



Hình 3-5: Điều chỉnh các loại file nhãn.

Kết quả cuối cùng, ta thu được tập dữ liệu bao gồm 3 thư mục: train, valid, test.

test	File folder					6/16/2024 1:44 AM
train	File folder					6/16/2024 1:44 AM
valid	File folder					6/16/2024 1:44 AM
data	Yaml Source File	1 KB	No	1 KB	57%	6/16/2024 1:44 AM
README.dataset	Text Source File	1 KB	No	1 KB	12%	6/16/2024 1:44 AM
README.roboflow	Text Source File	1 KB	No	2 KB	47%	6/16/2024 1:44 AM

Hình 3-6: Tập dữ liệu biên báo giao thông.

Lưu ý: file data.yaml rất quan trọng. Trong đó chứa các đường dẫn tới các tập dữ liệu train, valid, test, các lớp và số lượng lớp. Nên điều chỉnh các đường dẫn sao cho phù hợp để khi huấn luyện có thể trở đến được tập dữ liệu.

3.1.2. Quá trình huấn luyện mô hình:

Nhóm chúng em sử dụng mô hình YOLOv8 sử dụng phiên bản yolov8m.pt để tiến hành huấn luyện. Như bảng Bảng 2-1 đã nói, Phiên bản đuôi n cho ra độ chính xác của khá cao và lượng tham số ở mức thấp, thời gian huấn luyện cũng khá nhanh. Chúng em không sử dụng phiên bản sau vì do hạn chế về thiết bị, không có nhiều thời gian để huấn luyện

Chúng em sử dụng nền tảng Kaggle để huấn luyện với 2 bộ GPU T4. Các file mô hình, chúng em lấy từ bộ thư viện ultralytics. Bộ thư viện này chứa nhiều mô hình về YOLO và các câu lệnh để huấn luyện mô hình YOLO.

Chúng em sử dụng những file cấu hình mặc định của YOLO để huấn luyện. Tiến hành ghi đè lên file dữ liệu của YOLO thành file data.yaml của nhóm.

```
%cd /kaggle/working/

!yolo task=detect mode=train model=/kaggle/input/best-30-new-traffic-
sign/best_30_new.pt data=/kaggle/working/Traffic-Sign-at-HCMC-10/data.yaml
epochs=150 imgsz=640 plots=True patience = 0 device = 0,1
```

Chúng em tiến hành huấn luyện 100 epoch, thu được các kết quả và được phân tích ở CHƯƠNG 4:

yoioy8_custom_training

Draft saved

File

Edit

View

Run

Add-ons

Help

+

✂

📄

📁

▶

▶▶

Run All

Code ▾

● Draft Session off (run a cell to start)

Using 4 dataloader workers

Logging results to **runs/detect/train**

Starting training for 150 epochs...

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size		
1/150	3.97G	0.6269	0.4327	0.9978	7	640: 1		
	Class	Images	Instances	Box(P	R	mAP50	m	
	all	1971	2874	0.943	0.921	0.952		0.794

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size		
2/150	4.07G	0.6663	0.4654	1.019	9	640: 1		
	Class	Images	Instances	Box(P	R	mAP50	m	
	all	1971	2874	0.931	0.921	0.947		0.784

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size		
3/150	4.07G	0.7159	0.5258	1.037	8	640: 1		
	Class	Images	Instances	Box(P	R	mAP50	m	
	all	1971	2874	0.927	0.902	0.935		0.759

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size		
4/150	4.07G	0.7556	0.5881	1.062	15	640: 1		
	Class	Images	Instances	Box(P	R	mAP50	m	
	all	1971	2874	0.934	0.894	0.939		0.764

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size		
5/150	4.08G	0.7556	0.5807	1.059	11	640: 1		
	Class	Images	Instances	Box(P	R	mAP50	m	
	all	1971	2874	0.936	0.9	0.943		0.768

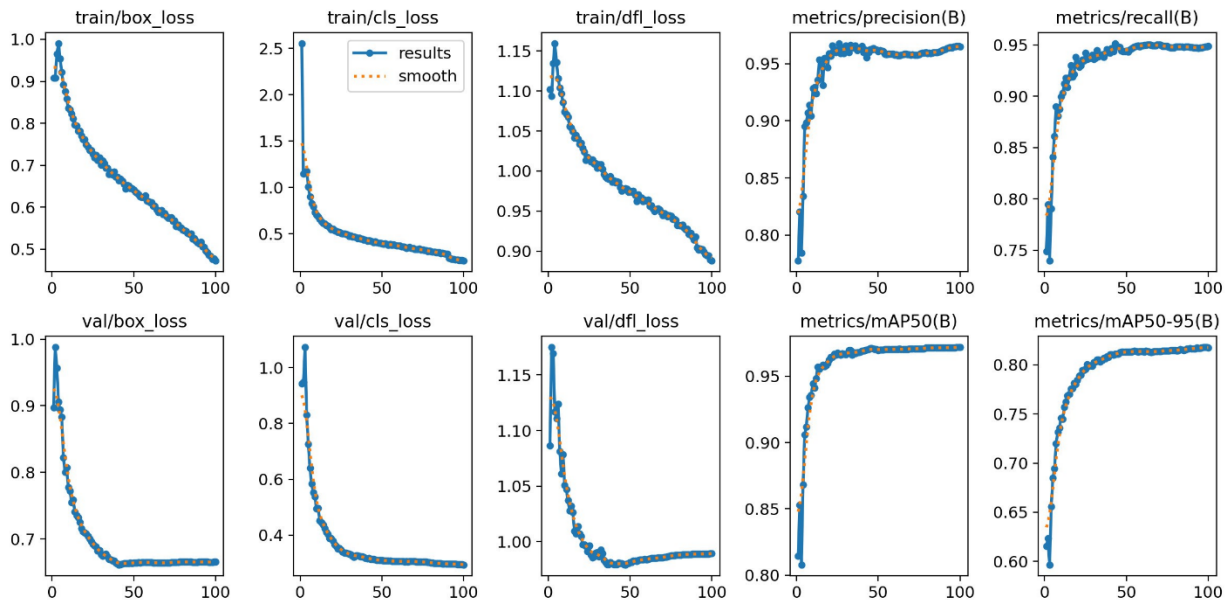
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size		
23/150	4.07G	0.6914	0.4852	1.019	18	640:		

Hình 3-7: Quá trình huấn luyện mô hình với 150 epochs.

CHƯƠNG 4: ĐÁNH GIÁ KẾT QUẢ VÀ KIỂM THỬ MÔ HÌNH.

4.1. KẾT QUẢ SAU KHI HUẤN LUYỆN MÔ HÌNH:

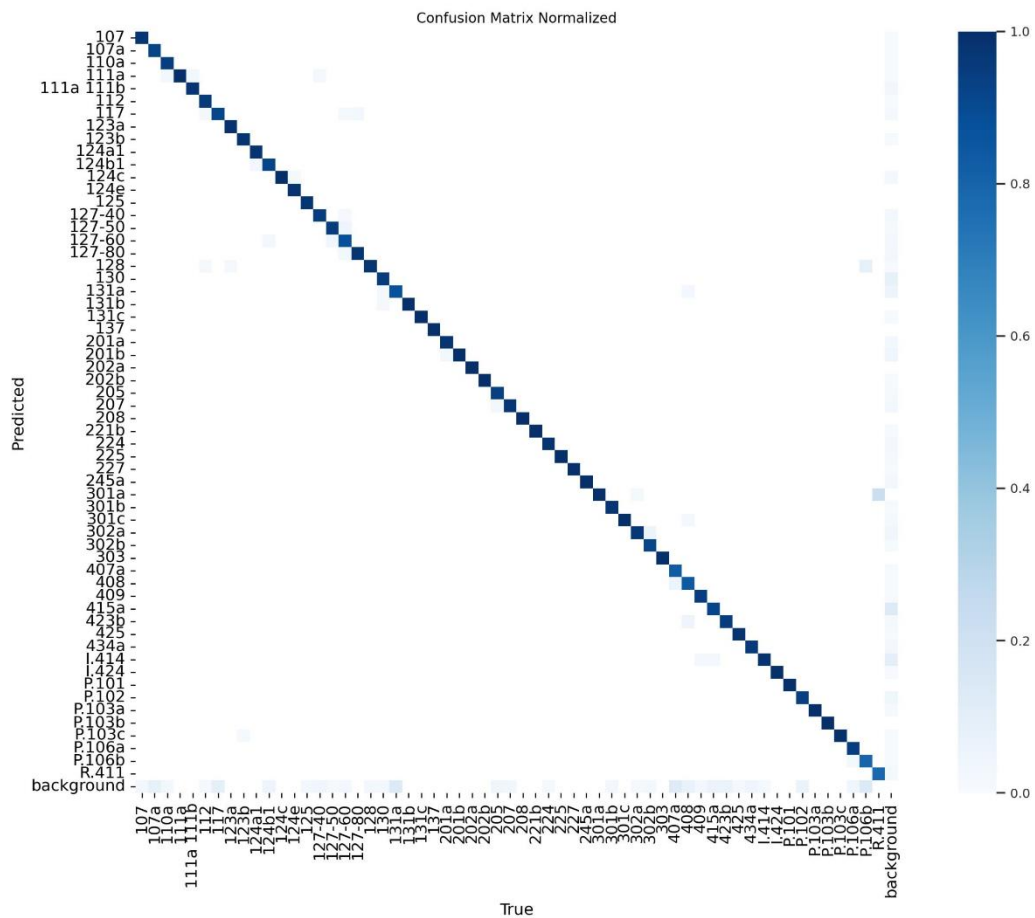
Đồ thị biểu diễn box_loss, class_loss, dft_loss, mAP50, mAP50-95, precision và recal khi huấn luyện mô hình qua 100 epochs.



Hình 4-1: Kết quả sau khi huấn luyện mô hình với số epoch là 100

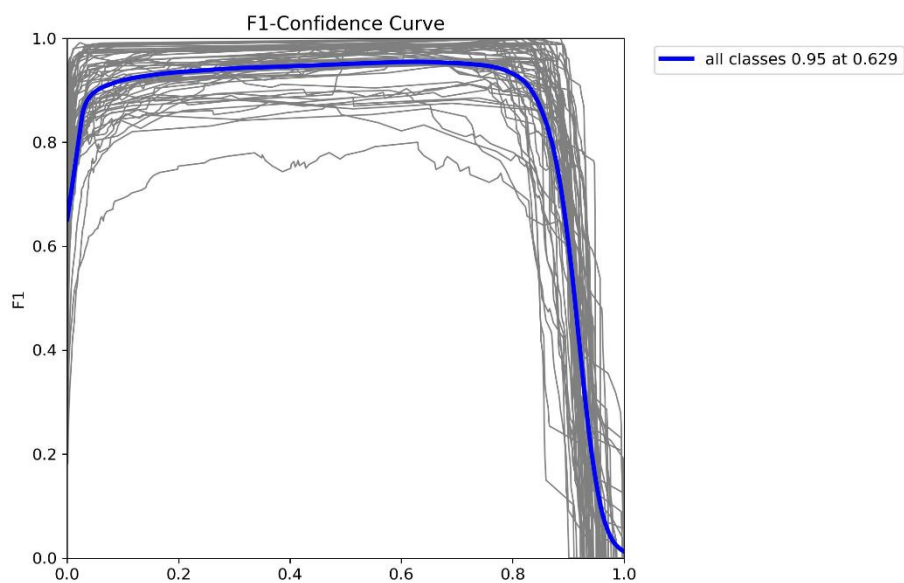
Quan sát kết quả, ta thấy các giá trị loss đang giảm mạnh đồng thời độ chính xác trung bình tăng lên đáng kể với mAP50 đạt hơn 0.95 và mAP50-95 đạt hơn 0.8 sau khi huấn luyện 100 epochs. Ngoài ra precision và recall cũng khá là cao đạt 0.95. Chứng tỏ rằng mô hình được huấn luyện một cách hiệu quả.

Quan sát confusion matrix đã chuẩn hóa: ta thấy hầu như các điểm True Positive đều đạt được mức khá cao (đường chéo đậm), nhưng cũng có một số lỗi mà mô hình đã xác định sai nhưng khá ít. Nói chung, mô hình đã huấn luyện này có thể xác định được các biến báo một cách chính xác.



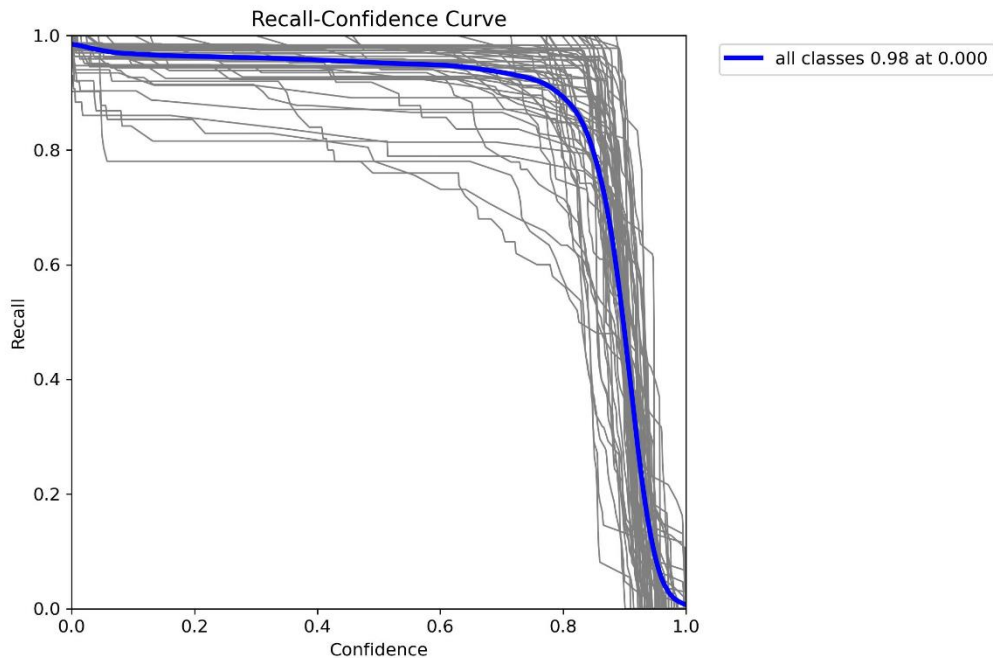
Hình 4-2: Confusion matrix.

Một số đường cong tham số khác:



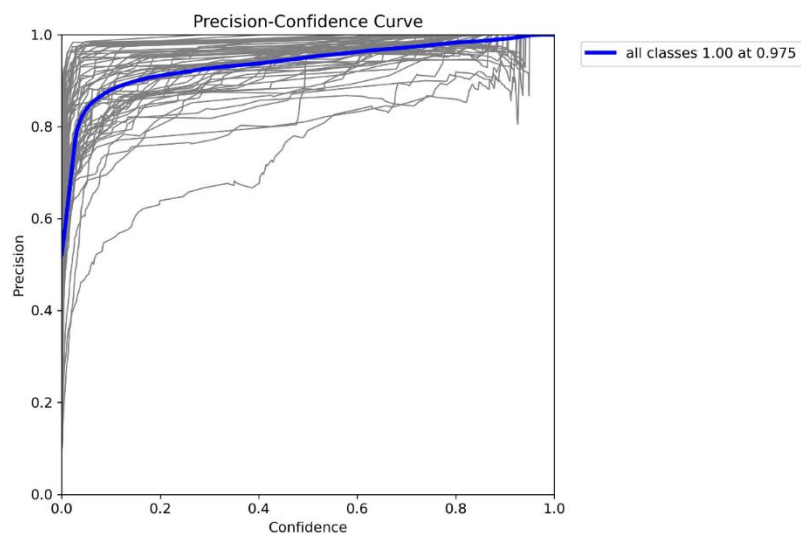
Hình 4-3: Đường cong F1-Confidence Curve.

Nhận xét: F1 score khá cao trong phạm vi rộng của các ngưỡng độ tin cậy và chỉ giảm mạnh khi độ tin cậy gần 1. Mô hình hoạt động tốt và cân bằng giữa precision và recall cho đến khi đạt ngưỡng độ tin cậy cao, nơi có thể trở nên quá tự tin.



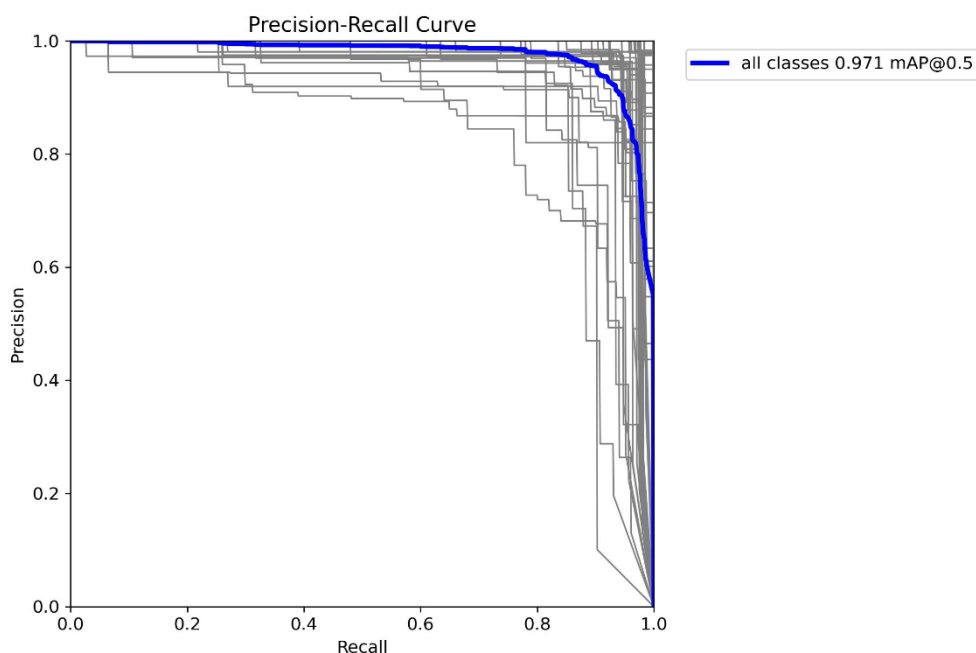
Hình 4-4: Đường cong Recall-Confidence Curve

Nhận xét: Recall duy trì ở mức cao trong phần lớn các ngưỡng độ tin cậy và chỉ bắt đầu giảm đáng kể khi độ tin cậy vượt qua 0.8.



Hình 4-5: Đường cong Precision-Confidence Curve

Nhận xét: Precision thấp ở các ngưỡng độ tin cậy thấp và tăng đáng kể khi ngưỡng độ tin cậy tăng, ổn định gần mức 1. Mô hình đạt precision cao ở các ngưỡng độ tin cậy cao, cho thấy khi mô hình rất tự tin về dự đoán của mình, nó hầu như luôn đúng. Tuy nhiên, ở các ngưỡng độ tin cậy thấp hơn, tỷ lệ false positives cao hơn.



Hình 4-6: Đường cong Precision-Recall Curve

Nhận xét: Đường cong màu xanh lam ở gần trên đỉnh cho thấy mô hình duy trì độ chính xác cao trên một loạt các giá trị recall, cho thấy nó hoạt động tốt trong việc dự đoán lớp dương tính.

Nhìn chung các kết quả đường cong cũng cho thấy được rằng mô hình đã được huấn luyện một cách hiệu quả và đạt được độ chính xác khá cao. Tùy thuộc vào ứng dụng, chúng ta có thể chọn ngưỡng độ tin cậy cân bằng giữa precision và recall một cách phù hợp.

Đánh giá kết quả trên tập validation

Bảng 4-1: Bảng kết quả xét trên tập validation

Class	Images	Instances	Box(P	R	mAP50	m
all	1985	2874	0.965	0.947	0.971	0.818
107	1985	45	0.989	0.978	0.993	0.885
107a	1985	39	1	0.916	0.971	0.783
110a	1985	38	0.958	0.947	0.97	0.867
111a	1985	42	0.933	1	0.995	0.789
111a 111b	1985	48	0.94	0.979	0.975	0.716
112	1985	66	0.997	0.97	0.984	0.849
117	1985	46	0.901	0.891	0.926	0.757
123a	1985	66	1	0.975	0.989	0.866
123b	1985	52	0.972	0.981	0.993	0.843
124a1	1985	38	0.992	0.974	0.994	0.864
124b1	1985	40	0.976	0.925	0.987	0.85
124c	1985	38	1	0.97	0.994	0.792
124e	1985	60	0.995	0.983	0.995	0.934
125	1985	43	1	0.936	0.979	0.816
127-40	1985	55	0.961	0.945	0.949	0.823
127-50	1985	57	0.964	0.947	0.928	0.853
127-60	1985	82	0.958	0.866	0.96	0.869
127-80	1985	47	0.979	0.972	0.987	0.924
128	1985	42	0.97	0.929	0.96	0.798
130	1985	77	0.862	0.896	0.906	0.698
131a	1985	62	0.888	0.871	0.88	0.681
131b	1985	38	0.971	1	0.995	0.919
131c	1985	45	0.975	1	0.995	0.786
137	1985	37	0.994	1	0.995	0.852
201a	1985	45	0.992	0.978	0.994	0.832
201b	1985	50	0.962	1	0.988	0.876
202a	1985	46	0.993	1	0.995	0.928
202b	1985	51	0.973	1	0.983	0.904
205	1985	49	1	0.943	0.973	0.767
207	1985	67	0.984	0.926	0.975	0.861
208	1985	50	0.995	1	0.995	0.905
221b	1985	42	0.987	1	0.995	0.912
224	1985	59	0.961	0.983	0.994	0.818
225	1985	51	0.962	1	0.995	0.729
227	1985	37	0.971	1	0.995	0.818
245a	1985	50	0.96	0.965	0.986	0.838
301a	1985	37	0.985	1	0.995	0.689

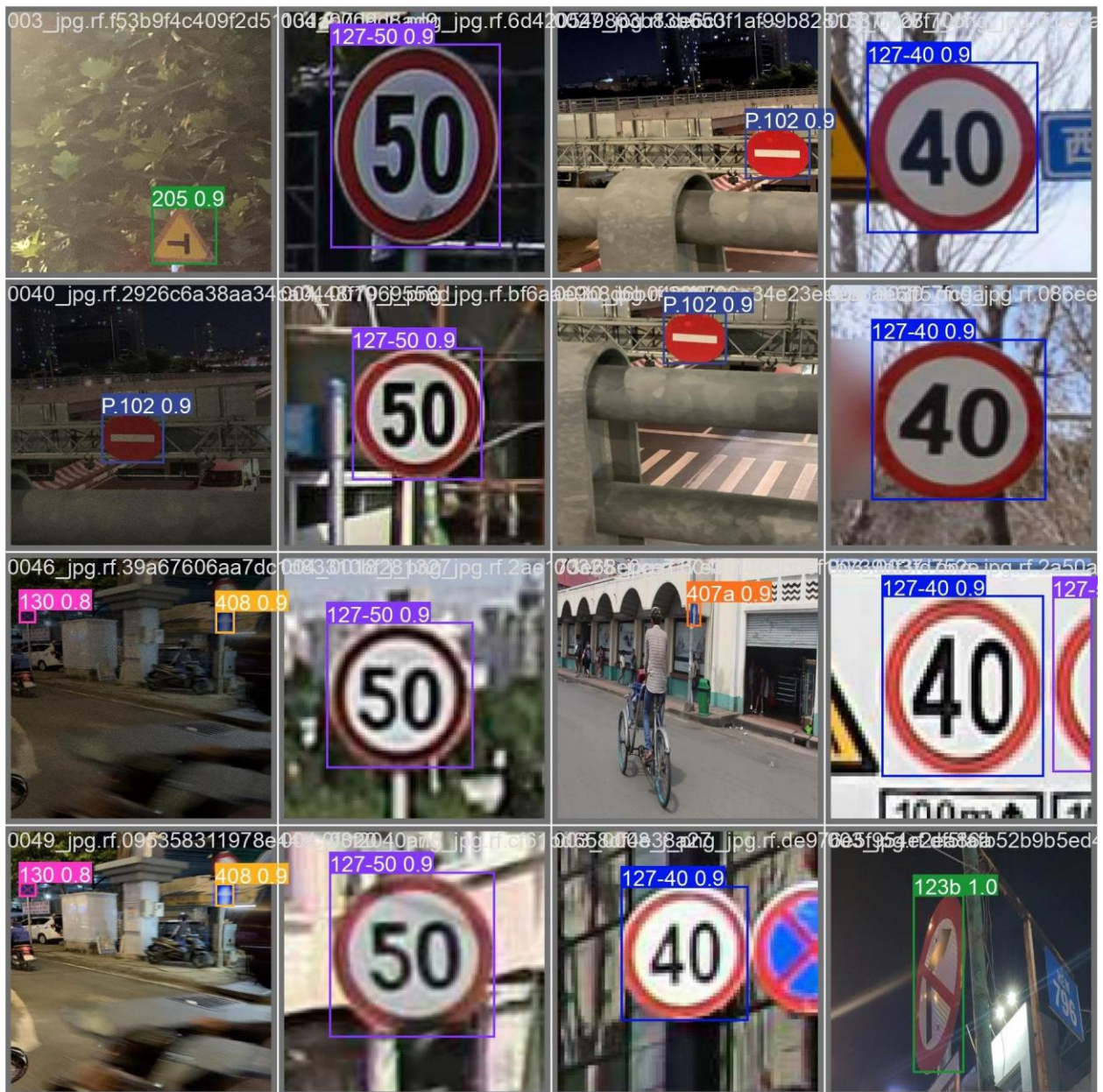
301b	1985	47	0.949	0.957	0.974	0.84
301c	1985	38	0.922	0.974	0.968	0.864
302a	1985	61	0.97	0.984	0.994	0.764
302b	1985	38	1	0.963	0.988	0.868
303	1985	37	0.992	1	0.995	0.839
407a	1985	41	0.954	0.732	0.884	0.63
408	1985	43	0.936	0.814	0.876	0.752
409	1985	39	0.997	0.923	0.983	0.815
415a	1985	50	0.843	0.76	0.862	0.692
423b	1985	69	0.98	0.928	0.983	0.801
425	1985	52	0.981	0.977	0.974	0.778
434a	1985	44	0.926	0.955	0.979	0.825
I.414	1985	63	0.836	0.952	0.925	0.782
I.424	1985	55	0.942	0.964	0.973	0.902
P.101	1985	54	0.996	1	0.995	0.748
P.102	1985	49	0.954	0.843	0.955	0.831
P.103a	1985	40	0.975	1	0.995	0.872
P.103b	1985	40	0.993	1	0.995	0.803
P.103c	1985	45	0.974	1	0.995	0.87
P.106a	1985	53	0.975	0.943	0.982	0.868
P.106b	1985	38	0.985	0.789	0.923	0.773
R.411	1985	41	0.962	0.78	0.943	0.673

Nhìn chung, khi xét từng biển báo các giá trị độ chính xác vẫn đảm bảo cho chúng ta có thể xác định các biển báo được.

Kiểm thử trên tập test: tiến hành dự đoán một hình. Kết quả ra chính xác với độ tin cậy cao đạt 0.91.



Hình 4-7: Hình minh họa cho việc dự đoán mô hình



Hình 4-8: Một ví dụ khác trong chạy thử mô hình.

4.2. TIẾN HÀNH CHẠY THỬ MÔ HÌNH SỬ DỤNG WEBCAM:

Chúng ta tiến hành sử dụng mô hình vừa huấn luyện và sử dụng webcam của máy tính như là đầu vào. Ở đây, chúng ta sử dụng opencv, một thư viện rất mạnh trong lĩnh vực thị giác máy tính. Nó giúp chúng ta có thể kết nối với webcam máy tính để lấy ra được những khung hình. Sau đó những khung hình sẽ được mô hình vừa huấn luyện xác định loại biển báo mà chúng ta đang chiếu.

Đoạn mã cho chức năng đó như sau:

```
import cv2
from ultralytics import YOLO

model = YOLO(
    r"D:\University\Nam 4\HK2\Nhap mon tri tue nhan tao\Model\Bao Cao\Lang1\best.pt")

cap = cv2.VideoCapture(0)

while cap.isOpened():
    ret, frame = cap.read()

    if not ret:
        break

    results = model.predict(frame, conf=0.5)

    # Vẽ bounding box

    for box in results[0].boxes:
        x1, y1, x2, y2 = map(int, box.xyxy[0].cpu().numpy())
        confident_score = float(box.conf.cpu().numpy())
        label_id = int(box.cls.cpu().numpy())
        label = model.names[label_id]

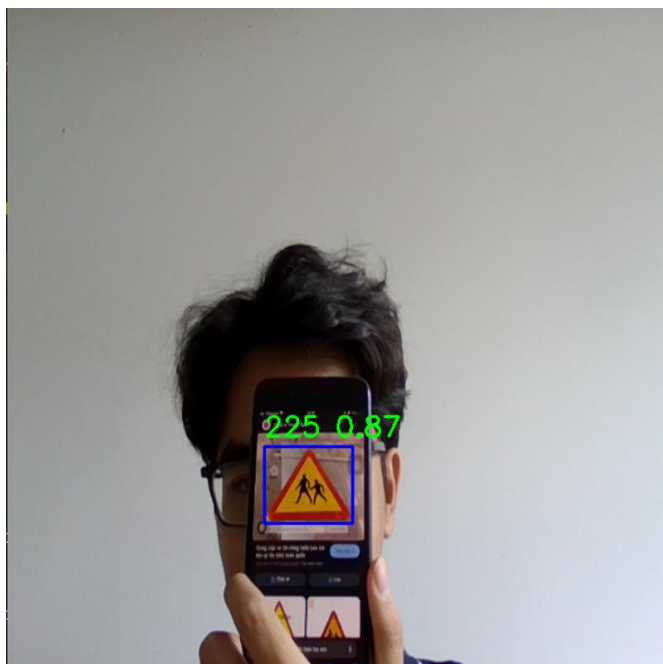
        cv2.rectangle(frame, (x1, y1), (x2, y2), (255, 0, 0), 2)
        cv2.putText(frame, f'{label} {confident_score:.2f}', (x1, y1 - 10),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0, 255, 0), 2)

    cv2.imshow("Video", frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```


Kết quả chạy đoạn mã: xác định chính xác loại biển báo.



Hình 4-9: Kết quả chạy mô phỏng đoạn mã.

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. KẾT LUẬN:

Qua đề tài “Ứng dụng mô hình mạng YOLO trong việc nhận dạng các biển báo giao thông trên đường ở Việt Nam” đã chứng minh hiệu quả của mô hình mạng YOLO trong việc nhận dạng biển báo giao thông với tốc độ xử lý nhanh chóng và độ chính xác cao. Bằng cách xử dụng các biển báo giao thông đặc trưng của Việt Nam, mô hình đã được tối ưu hoá, đảm bảo nhận diện đúng các biển báo trong điều kiện giao thông phức tạp của Việt Nam. Điều này cho thấy khả năng ứng dụng của YOLO trong các hệ thống giao thông thông minh và các thiết bị di động là rất tiềm năng trong việc nâng cao an toàn giao thông. Bằng việc tích hợp mô hình YOLO vào các hệ thống giám sát giao thông, chúng ta có thể tự động hóa quá trình nhận dạng biển báo, giúp giảm bớt gánh nặng cho nhân viên và tăng cường sự chính xác so với phương pháp truyền thống. Điều này có thể mang lại lợi ích lớn cho các thành phố và khu vực đông dân cư, nơi mà lưu lượng giao thông cao.

Tuy nhiên, để ứng dụng mô hình YOLO một cách hiệu quả và phát triển, cần liên tục cải thiện, cập nhật liên tục dữ liệu và tối ưu hoá mô hình để nâng cao và duy trì độ chính xác và hiệu suất. Tiếp theo đó, chúng ta cần triển khai mô hình trong môi trường thực tế, phát triển các ứng dụng di động và hợp tác với các tổ chức giao thông để tạo ra các hệ thống hỗ trợ lái xe và giám sát giao thông.

5.2. HƯỚNG PHÁT TRIỂN:

Trong tương lai, việc phát triển và ứng dụng mô hình mạng YOLO trong việc nhận dạng các biển báo giao thông trên đường ở Việt Nam có thể tiếp tục mở ra nhiều cơ hội và tiềm năng như việc tối ưu hóa mô hình YOLO cho điều kiện địa phương. Để đảm bảo tính chính xác và hiệu suất cao khi nhận dạng biển báo giao thông trong môi trường đường ở Việt Nam, cần tiến hành tối ưu hóa mô hình YOLO cho các đặc điểm địa phương như điều kiện ánh sáng, phong cảnh đường, và loại biển báo cụ thể. Bên cạnh đó, ta có thể phát triển dịch vụ và ứng dụng tiện ích như việc tích hợp mô hình YOLO vào các ứng dụng di động hoặc các hệ thống dẫn đường có thể cung cấp thông tin về biển báo giao thông một cách

nhANH chóng và thuận tiện cho người dùng. Điều này có thể giúp tăng cường sự an toàn và sự thuận tiện khi di chuyển trên đường.

Ngoài ra, đề tài trên còn có thể nghiên cứu và áp dụng trong lĩnh vực xe tự lái. Sự phát triển của công nghệ xe tự lái đang ngày càng trở nên quan trọng, và việc sử dụng mô hình YOLO để nhận dạng biển báo giao thông có thể là một phần quan trọng trong hệ thống điều khiển và giám sát của các phương tiện tự hành. Điều này có thể đóng vai trò quan trọng trong việc tạo ra một môi trường giao thông an toàn và hiệu quả. Việc ứng dụng mô hình mạng YOLO trong việc nhận dạng các biển báo giao thông trên đường ở Việt Nam có thể đem lại nhiều lợi ích về mặt an toàn, tiện ích và hiệu suất giao thông cho cộng đồng.

TÀI LIỆU THAM KHẢO

- [1] What is a neural network? | Types of neural networks. (n.d.). Cloudflare. Retrieved June 17, 2024, from <https://www.cloudflare.com/learning/ai/what-is-neural-network/>
- [2] Các hàm kích hoạt (activation function) trong neural network. (2019, September 23). Viet-Anh on Software. Retrieved June 17, 2024, from <https://www.vietanh.dev/blog/2019-09-23-cac-ham-kich-hoat-activation-function-trong-neural-networks>
- [3] SHARMA, S. (n.d.). Activation Functions in Neural Networks | by SAGAR SHARMA. Towards Data Science. Retrieved June 17, 2024, from <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>
- [4] Jain, S. (2024, May 16). Backpropagation in Neural Network. GeeksforGeeks. Retrieved June 17, 2024, from <https://www.geeksforgeeks.org/backpropagation-in-neural-network/>
- [5] Mirkhan, A. (n.d.). YOLO Algorithm: Real-Time Object Detection from A to Z. Kili Technology. Retrieved June 17, 2024, from <https://kili-technology.com/data-labeling/machine-learning/yolo-algorithm-real-time-object-detection-from-a-to-z>
- [6] Jocher, G. (2023, November 12). YOLOv8 - Ultralytics YOLO Docs. Ultralytics YOLOv8. Retrieved June 18, 2024, from <https://docs.ultralytics.com/models/yolov8/#usage-examples>
- [7] Malviya, N. (2023, August 5). Object Detection — Anchor Box VS Bounding Box | by Nikita Malviya. Medium. Retrieved June 17, 2024, from

<https://medium.com/@nikitamalviya/object-detection-anchor-box-vs-bounding-box-bf1261f98f12>

- [8] Prathap, P. (2023, July 28). The Secret to Understanding CNNs: Convolution, Feature Maps, Pooling and Fully Connected Layers! Medium. Retrieved June 17, 2024, from <https://medium.com/@prajeeshprathap/the-secret-to-understanding-cnns-convolution-feature-maps-pooling-and-fully-connected-layers-97055431a847>
- [9] Accuracy vs. precision vs. recall in machine learning: what's the difference? (n.d.). Evidently AI. Retrieved June 17, 2024, from <https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall>
- [10] loss - YOLOv8 dfl_loss metric. (2023, April 6). Stack Overflow. Retrieved June 17, 2024, from <https://stackoverflow.com/questions/75950283/yolov8-dfl-loss-metric>
- [11] Noyes, L. (2023, December 17). What is the difference between mAP50 and mAP50-95? | 4 Answers from Research papers. Typeset.io. Retrieved June 17, 2024, from <https://typeset.io/questions/what-is-the-difference-between-map50-and-map50-95-5asxtut1cj>
- [12] (n.d.). colab.google. Retrieved June 17, 2024, from <https://colab.google/>
- [13] What is roboflow? | 4 Answers from Research papers. (2023, November 10). Typeset.io. Retrieved June 17, 2024, from <https://typeset.io/questions/what-is-roboflow-28xw6lpv52>

PHỤ LỤC

Biên bản cuộc họp:

Cuộc họp đầu tiên được tổ chức vào lúc 19h00 ngày 22 tháng 05 năm 2024 với sự có mặt đầy đủ của tất cả các thành viên trong nhóm.

Nội dung: Lên kế hoạch, lập danh sách ý tưởng cho nội dung tìm hiểu. Phân công nhiệm vụ cho từng bạn.

Kết quả: Bảng phân công công việc (mục 3), lộ trình thực hiện đồ án.

Lộ trình thực hiện đồ án như sau:

Từ ngày 22 đến ngày 03/06/2024: thu thập dữ liệu biến bảo, tiến hành gắn nhãn (label) cho dữ liệu.

Từ ngày 04 đến ngày 15/06/2024: tiến hành training với data đã gắn nhãn → lưu trữ model đã train.

Từ ngày 16 đến ngày 20/06/2024: tiến hành kiểm thử mô hình (kiểm tra độ chính xác), phân tích các kết quả, đồ thị, và một số công việc liên quan khác, tiến hành viết báo cáo và slide để thuyết trình.

Trưởng nhóm ký tên

Dũng

Nguyễn Dũng

Lộ trình thực hiện:

Nhìn chung đạt đúng tiến độ yêu cầu, nhưng có số mục như là database chưa đạt đúng thời gian đã đặt ra, còn phải cân nhắc nhờ

Phản huấn luyện đảm bảo đúng tiến độ, chính xác

Kết quả hoàn thành vào ngày 19/06/2024, Sau ngày đó, có một số cái cần sửa và nâng cấp.

Bảng phân công công việc:

Kế hoạch thực hiện, phân chia nhiệm vụ cho các thành viên:

STT	Thành viên	Chức năng	Nhiệm vụ	Mức độ hoàn thành, đóng góp	Trung bình	Nhận xét	Ký tên
1	Nguyễn Dũng	Trưởng nhóm	Phân công nhiệm vụ, tổng hợp ý kiến và triển khai công việc cho các thành viên, Huấn luyện và kiểm thử mô hình (năm	Tự đánh giá: 100%	99.9%	Tự nhận xét: Đã hoàn thành tốt vai trò nhóm trưởng	
				99%		Có tham gia quá trình xây dựng kế hoạch nhóm...	
				100%		Đóng góp nhiều về train mô hình, tích cực xây dựng kế hoạch nhóm,	

			chính); Viết word.			đóng góp tích cực.	
				100%		Tham gia tích cực tất cả các quá trình của nhóm, đưa ra kế hoạch cụ thể và giúp đỡ các thành viên còn lại trong nhóm. Thúc đẩy tiến trình cả nhóm hoàn thành nhanh.	
				100%		Hoàn thành tốt công việc của bản thân và giúp đỡ các thành viên khác. Phân chia công việc rõ ràng và công bằng	
				100%		Tham gia chủ động vào mọi hoạt động của nhóm, lập kế hoạch chi tiết rõ ràng, hỗ trợ các thành viên trong nhóm.	

				100%		Hoàn thành tốt vai trò	
2	Trần Quang Đài	Thành viên	Huấn luyện mô hình và kiểm thử.	Tự đánh giá: 85%	87.9%	Tự nhận xét: Có cố gắng train, có chuẩn bị nhưng còn 1 số lỗi. Dẫn đến chưa có kết quả mong muốn.	
				85%		Có tham gia vào quá trình huấn luyện mô hình, có chủ động, nhưng chưa có được kết quả mong muốn	
				95%		Có tham gia huấn luyện mô hình.	
				95%		Có tích cực tìm hiểu về train mô hình, tham gia đóng góp vào training.	
				85%		Có chủ động trong công việc, nhưng chưa đạt	

						kết quả khả quan.	
				85%		Có tìm hiểu và thường xuyên tương tác với nhóm, nhưng khi tham gia mô hình còn lỗi và chưa có kết quả tốt nhất	
				85%		Có chủ động trong công việc.	
3	Đỗ Quang Hiếu	Thành viên	Thu thập dữ liệu, đi chụp hình các biển báo trên đường, gắn nhãn dữ liệu, Làm PPT	Tự đánh giá: 88%	88.4%	Tự nhận xét: Tham gia tích cực vào phần thu thập dữ liệu, tham gia quá trình gắn label, chưa chuẩn bị tốt phần vẽ ma trận confusion, ROC curve.	
				90%		Có chủ động, thu thập dữ liệu nhưng còn phải nhắc nhở, gắn nhãn còn có một số chỗ bị sai (vấn đề là kiểu có một số biển	

					tận 2 class nên khi dán mỗi thứ 1 nếu có 1 số class nó không đạt được cái ngược màu xanh để train nhận kết quả như mong muốn), nhưng có sự khắc phục	
				90%	Có đóng góp vào xây dựng PPT	
				88%	Có làm nhiệt tình nhưng còn sai sót, cần được nhắc nhở nhưng đã có điều chỉnh	
				88%	Gán nhãn còn sai sót nhưng đã kịp thời điều chỉnh, thu thập được nhiều dữ liệu biến báo.	
				90%	Tham gia tích cực vào các nhiệm vụ được giao.	

				85%		Còn bị động trong công việc của nhóm.	
4	Đặng Ngọc Hoàng	Thành viên	Thu thập dữ liệu, đi chụp hình các biển báo trên đường, gắn nhãn dữ liệu	Tự đánh giá: 87%	87%	Tự nhận xét: Có dán nhãn sai nhưng chủ động tương tác và khắc phục, thu thập dữ liệu nhưng còn ít.	
				87%		Có thu thập dữ liệu nhưng còn phải nhắc nhở, khá ít, thụ động, gắn nhãn còn có một số chỗ bị sai (vấn đề là kiểu có một số biển tận 2 class nên khi dán mỗi thứ 1 nếu có 1 số class nó không đạt được cái ngược màu xanh để train nhận kết quả như mong muốn), nhưng có sự khắc phục	
				90%		Có chủ động gắn nhãn phân loại	

				88%		Đóng góp thu thập dữ liệu tích cực, nhưng hơi thụ động.	
				85%		Gắn nhãn còn sai sót, nhưng đã khắc phục.	
				87%		Có chủ động nhắn tin tương tác với nhóm trưởng khi gắn nhãn sai và kịp thời khắc phục nhanh chóng, có thu thập dữ liệu.	
				85%		Đóng góp thu thập dữ liệu, dán nhãn tích cực.	
5	Trần Bích Ngọc	Thành viên	Thu thập dữ liệu, đi chụp hình các biển báo trên đường, gắn nhãn dữ liệu, Viết word	Tự đánh giá: 90%	90%	Tự nhận xét: Tham gia vào dán, thu thập dữ liệu và viết word.	
				90%		Có chủ động thu thập dữ liệu nhưng khá ít, gắn nhãn cũng tương đối chính xác. Khắc phục được nhiều làm	

						sai nhanh chóng nhưng khá thụ động	
				90%		Có chủ động tìm hiểu về data	
				90%		Thu thập, đồng góp dữ liệu, tham gia gán label.	
				90%		Có chủ động thu thập dữ liệu nhưng còn ít, dẫn nhãn nhanh và chính xác.	
				90%		Hoàn thành ổn công việc gán nhãn và thu thập dữ liệu mặc dù còn hơi ít.	
				90%		Chủ động sửa chữa những sai sót nhanh chóng	
6	Lạc Hải Sơn	Thành viên	Kiểm thử mô hình, Thu thập dữ liệu.	Tự đánh giá: 85%	82.9%	Tự nhận xét Chưa tham gia nhiều vào quá trình xây dựng đồ án	
				85%		Chỉ tham gia vào quá trình	

						kiểm thử mô hình, có tham gia vào quá trình thu thập dữ liệu	
				85%		Có tham gia vào quá trình kiểm thử mô hình, hơi ít tham gia vào quá trình hoạt động, còn thụ động và ít tương tác.	
				85%		Có tham gia vào quá trình xây dựng đồ án nhưng còn ít tương tác và thụ động	
				85%		Ít tương tác với các thành viên trong nhóm dẫn đến việc tham gia vào quá trình xây dựng đồ án còn ít	
				85%		Ít tương tác với nhóm, ít tham gia vào quá trình xây dựng đồ án.	

				70%		Ít tham gia các hoạt động trong bài báo cáo	
7	Trần Thuận Thiên	Thành viên	Gắn nhãn mô hình.	Tự đánh giá: 70%	82.1%	Tự nhận xét: Chưa hoàn thành tốt chuẩn bị tài liệu data	
				85%		Tham gia vào quá trình dán nhãn, dán nhãn dán nhãn cũng có sai những đã kịp thời sửa.	
				85%		Tích cực tham gia vào quá trình hoạt động nhóm, có tìm hiểu về data, gán label, có tích cực, nhưng chưa hoàn thành tốt chuẩn bị tài liệu về data.	
				85%		Tham gia đóng góp ý kiến trong nhóm thường xuyên, gắn nhãn tương đối chính xác nhưng chưa	

						thu thập dữ liệu nhiều.	
				85%		Tương tác nhiều với nhóm và cũng đưa ra được nhiều phương pháp. Gán nhãn còn sai sót ít nhưng chưa đóng góp nhiều vào thu thập dữ liệu.	
				85%		Tích cực tương tác với các thành viên trong nhóm và đóng góp nhiều ý kiến cho nhóm.	
				80%		Có chủ động tham luận nhưng chưa nghiêm túc trong công việc.	

Đã được thống nhất giữa các thành viên.

Chữ ký trưởng nhóm

Dũng

Nguyễn Dũng

Tự đánh giá kỹ năng và kiến thức

Nhóm tự đánh giá tổng hợp các kỹ năng, kiến thức đạt được qua đồ án môn học.

STT	Các kỹ năng	Đánh giá (*)
1	Kỹ năng làm việc nhóm, giao tiếp, hành xử chuyên nghiệp, khả năng lãnh đạo và làm việc độc lập.	A
2	Kỹ năng tư duy phản biện.	B
3	Kỹ năng thuyết trình.	A
4	Giao tiếp kỹ thuật (viết báo cáo kỹ thuật)	A
5	Kỹ năng tư duy sáng tạo.	B
6	Kỹ năng quản lý dự án/ thời gian thực hiện dự án.	B
7	Hình thành nội dung, xác định vấn đề và kỹ năng giải quyết vấn đề.	A
8	Kiến thức, thực nghiệm qua đồ án môn học.	A

(*) Ghi chú: Đánh giá theo mức A/B/C/D (A: Rất tốt, B: Tốt, C: Trung bình, D: Chưa tốt)