**TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**

**KHOA CÔNG NGHỆ THÔNG TIN**



**TIỂU LUẬN NHẬN DẠNG ĐỐI TƯỢNG**

Ứng Dụng YOLOv8 Phân Loại Xe

Để Giám Sát Trong Trạm Thu Phí

**Giảng viên hướng dẫn: TRẦN CHÂU THANH THIỆN**

**Sinh viên thực hiện: VÕ NGUYỄN BẢO PHƯỚC**

**MSSV: 2200008333**

**Khoá: 2022 - 2026**

**Ngành/ chuyên ngành: TRÍ TUỆ NHÂN TẠO**

Tp HCM, tháng 4 năm 2025

**TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**

**KHOA CÔNG NGHỆ THÔNG TIN**



**TIỂU LUẬN NHẬN DẠNG ĐỐI TƯỢNG**

Ứng Dụng YOLOv8 Phân Loại Xe

Để Giám Sát Trong Trạm Thu Phí

**Giảng viên hướng dẫn: TRẦN CHÂU THANH THIỆN**

**Sinh viên thực hiện: VÕ NGUYỄN BẢO PHƯỚC**

**MSSV: 2200008333**

**Khoá: 2022 - 2026**

**Ngành/ chuyên ngành: TRÍ TUỆ NHÂN TẠO**

TPHCM, tháng 4 năm 2025

LỜI MỞ ĐẦU

Trong bối cảnh giao thông ngày càng phát triển và nhu cầu tự động hóa các hệ thống quản lý phương tiện ngày càng cao, việc ứng dụng trí tuệ nhân tạo vào giám sát và phân loại xe tại các trạm thu phí trở thành một hướng đi thiết thực và đầy tiềm năng. Các phương pháp truyền thống như nhận dạng thủ công hoặc dựa vào cảm biến vật lý thường tồn tại nhiều hạn chế về chi phí, độ chính xác và khả năng mở rộng.

Với sự tiến bộ mạnh mẽ của thị giác máy tính (Computer Vision), các mô hình học sâu như YOLO (You Only Look Once) đã chứng minh được hiệu quả vượt trội trong việc phát hiện và phân loại đối tượng theo thời gian thực. Trong đó, YOLOv8, phiên bản mới nhất của dòng mô hình YOLO, mang đến nhiều cải tiến về tốc độ và độ chính xác, đặc biệt phù hợp với các ứng dụng yêu cầu xử lý nhanh như giám sát giao thông.

Đề tài này tập trung nghiên cứu và triển khai ứng dụng YOLOv8 vào việc phát hiện và phân loại các loại xe (xe máy, ô tô con, xe tải, xe buýt, v.v...) tại trạm thu phí, nhằm phục vụ cho mục đích giám sát, thống kê và hỗ trợ thu phí tự động. Việc áp dụng mô hình này không chỉ nâng cao hiệu quả quản lý mà còn góp phần thúc đẩy quá trình chuyển đổi số trong lĩnh vực giao thông vận tải.

LỜI CẢM ƠN

Trước tiên, em xin gửi lời cảm ơn chân thành đến Ban Giám Hiệu và các thầy cô trường Đại học Nguyễn Tất Thành, đặc biệt là Khoa Công nghệ thông tin, đã tạo điều kiện và cung cấp kiến thức trong suốt quá trình học tập và thực hiện đề tài.

Em xin gửi lời biết ơn sâu sắc đến thầy Trần Châu Thanh Thiện đã hết lòng hướng dẫn, góp ý và định hướng để em có thể hoàn thành tốt nhất đề tài này.

Em xin chân thành cảm ơn gia đình và bạn bè đã động viên, hỗ trợ em trong quá trình thực hiện đề tài. Đặc biệt, em xin cảm ơn những ý kiến đóng góp quý báu từ các thầy cô và đồng nghiệp trong suốt quá trình nghiên cứu.

Cuối cùng, em xin gửi lời cảm ơn đến tất cả những ai đã gián tiếp hoặc trực tiếp đồng hành cùng em trong suốt hành trình thực hiện đề tài này.

TPHCM, tháng 4 năm 2025

Họ tên sinh viên thực hiệu

(ký, ghi rõ họ tên)

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH  **TRUNG TÂM KHẢO THÍ** | **KỲ THI KẾT THÚC HỌC PHẦN**  **HỌC KỲ 2 NĂM HỌC 2024 – 2025** |

**PHIẾU CHẤM THI TIỂU LUẬN**

BM-ChT-11

Môn thi: Nhận Dạng Đối Tượng Lớp học phần:22DTH3C

Nhóm sinh viên thực hiện :Nhóm 1

1.Nguyễn Đoàn Thanh Thảo Tham gia đóng góp: ALL

2.Võ Nguyễn Bảo Phước Tham gia đóng góp: ALL

3. Võ Trần Duy Tân Tham gia đóng góp: ALL

Ngày thi: 24/12/2024 Phòng thi:L.608

Đề tài tiểu luận: Ứng Dụng YOLOv8 Phân Loại Xe Để Giám Sát Trong Trạm Thu Phí

Phần đánh giá của giảng viên (căn cứ trên thang rubrics của môn học):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tiêu chí (theo CĐR HP)** | **Đánh giá của GV** | | **Điểm tối đa** | **Điểm đạt được** |  |
| Cấu trúc của báo cáo |  | |  |  |  |
| Nội dung |  | |  |  |  |
| * Các nội dung thành phần |  | |  |  |  |
| * Lập luận |  | |  |  |  |
| * Kết luận |  | |  |  |  |
| Trình bày |  | |  |  |  |
| **TỔNG ĐIỂM** |  | |  |  |  |
|  | | **Giảng viên chấm thi**  *(ký, ghi rõ họ tên)* | | | |

MỤC LỤC

[LỜI MỞ ĐẦU 1](#_Toc196035585)

[LỜI CẢM ƠN 2](#_Toc196035586)

[MỤC LỤC 4](#_Toc196035587)

[Chương 1: Giới thiệu đề tài 7](#_Toc196035588)

[1.1 Đặt vấn đề 7](#_Toc196035589)

[1.2 Vấn đề nghiên cứu 7](#_Toc196035591)

[1.3 Đối tượng nghiên cứu 2](#_Toc196035592)

[1.4 Phạm vi nghiên cứu 2](#_Toc196035593)

[1.5 Lý do chọn đề tài 2](#_Toc196035594)

[Chương 2: Cơ sở lý thuyết 4](#_Toc196035595)

[2.1. Phát hiện đối tượng là gì? 4](#_Toc196035596)

[2.2 Các loại mô hình phát hiện đối tượng 5](#_Toc196035598)

[2.2.1 Các mô hình sử dụng một giai đoạn 5](#_Toc196035600)

[2.2.2 Các mô hình sử dụng hai giai đoạn 5](#_Toc196035601)

[2.3 Mạng YOLO You Only Look Once 6](#_Toc196035604)

[2.3.1 Giới thiệu về YOLO 6](#_Toc196035605)

[2.3.2 Các phiên bản YOLO 7](#_Toc196035607)

[2.3.3 Ứng dụng YOLO 8](#_Toc196035608)

[2.3.4 YOLOv1 9](#_Toc196035609)

[2.3.5 YOLOv8 16](#_Toc196035625)

[2.4 Tổng quan về bài toán phân loại xe để lập kế hoạch bảo trì mặt đường 18](#_Toc196035628)

[2.4.1 Ứng dụng Phát hiện đối tượng vào bài toán 18](#_Toc196035629)

[2.4.2 Thách thức và định hướng nghiên cứu 19](#_Toc196035630)

[Chương 3: Cơ sở thực tiễn 20](#_Toc196035631)

[3.1. Thực trạng hạ tầng giao thông tại Việt Nam 20](#_Toc196035632)

[3.2. Thách thức trong công tác giám sát và bảo trì hiện nay 20](#_Toc196035633)

[3.3. Vai trò của công nghệ Deep Learning trong giám sát giao thông 20](#_Toc196035634)

[3.4. Giải pháp đề xuất từ đề tài 20](#_Toc196035635)

[3.4.1. Mô hình tổng quát 21](#_Toc196035636)

[3.4.2. Công cụ và thư viện sử dụng 21](#_Toc196035637)

[3.4.3. Tiền xử lý video đầu vào 21](#_Toc196035638)

[3.4.4. Nhận dạng đối tượng với YOLOv8 21](#_Toc196035640)

[3.4.5. Tính toán điểm hư hại, tránh để trùng 22](#_Toc196035642)

[Chương 4: Thực nghiệm 23](#_Toc196035644)

[4.1 Sử dụng YOLOv8 trong bài toán phân loại xe để lập kế hoạch bảo trì mặt đường 23](#_Toc196035645)

[4.1.1 Lý do lựa chọn mô hình YOLOv8 23](#_Toc196035646)

[4.1.2 Ứng dụng trong bài toán phân loại xe để lập kế hoạch bảo trì mặt đường 23](#_Toc196035647)

[4.1.3 Cách thức hoạt động của mô hình YOLOv8 23](#_Toc196035648)

[4.2 kết quả và thay đổi để tối ưu bài toán. 24](#_Toc196035649)

[4.2.1 Lần 1: Tạo vùng ảnh hưởng 24](#_Toc196035651)

[4.2.2 Lần 2: Dùng thuật toán IOU 26](#_Toc196035656)

[Chương 5: Kết luận 28](#_Toc196035661)

[5.1 Nội dung đã thực hiện 28](#_Toc196035662)

[5.2 Hạn chế 28](#_Toc196035663)

[5.3 Hướng phát triển 28](#_Toc196035664)

[TÀI LIỆU THAM KHẢO 29](#_Toc196035665)

DANH MỤC HÌNH

[Hình 1.1 1](#_Toc196038172)

[Hình 2.1 4](#_Toc196038173)

[Hình 2.2 5](#_Toc196038174)

[Hinh 2.3 6](#_Toc196038175)

[Hình 2.4 6](#_Toc196038176)

[Hình 2.5 7](#_Toc196038177)

[Hình 2.6 9](#_Toc196038178)

[Hình 2.7 10](#_Toc196038179)

[Hình 2.8 11](#_Toc196038180)

[Hình 2.9 11](#_Toc196038181)

[Hình 2.10 11](#_Toc196038182)

[Hình 2.11 12](#_Toc196038183)

[Hình 2.12 12](#_Toc196038184)

[Hình 2.13 13](#_Toc196038185)

[Hình 2.14 13](#_Toc196038186)

[Hình 2.15 14](#_Toc196038187)

[Hình 2.16 14](#_Toc196038188)

[Hình 2.17 14](#_Toc196038189)

[Hình 2.18 15](#_Toc196038190)

[Hình 2.19 16](#_Toc196038191)

[Hình 2.20 16](#_Toc196038192)

[Hình 2.21 17](#_Toc196038193)

[Hình 2.22 18](#_Toc196038194)

[Hình 3.1 21](#_Toc196038195)

[Hình 3.2 22](#_Toc196038196)

[Hình 3.3 22](#_Toc196038197)

[Hình 4.1 24](#_Toc196038198)

[Hình 4.2 25](#_Toc196038199)

[Hình 4.3 25](#_Toc196038200)

[Hình 4.4 25](#_Toc196038201)

[Hình 4.5 25](#_Toc196038202)

[Hình 4.6 26](#_Toc196038203)

[Hình 4.7 26](#_Toc196038204)

[Hình 4.8 26](#_Toc196038205)

[Hình 4.9 27](#_Toc196038206)

Chương 1: Giới thiệu đề tài

1.1 Đặt vấn đề

**Làm thế nào để giám sát và phân loại phương tiện tại trạm thu phí một cách chính xác, hiệu quả và tiết kiệm?**

Đây là một trong những thách thức lớn đặt ra trong bối cảnh ngành giao thông vận tải Việt Nam đang chuyển mình mạnh mẽ theo hướng hiện đại hóa và số hóa.

Việt Nam hiện là một trong những quốc gia có tốc độ phát triển kinh tế nhanh ở khu vực Đông Nam Á. Tính đến năm 2024, tăng trưởng GDP đạt 7,09% – mức cao so với khu vực. Trong tiến trình phát triển đó, hạ tầng giao thông đóng vai trò thiết yếu, đặc biệt là hệ thống đường cao tốc và các trạm thu phí, vốn đang ngày càng mở rộng để đáp ứng nhu cầu lưu thông hàng hóa và hành khách.

Tính đến năm 2024, Việt Nam có hơn 2.000 km đường cao tốc và hàng trăm trạm thu phí BOT hoạt động. Dự kiến đến năm 2030, con số này sẽ còn tăng mạnh nhằm đáp ứng chiến lược phát triển giao thông quốc gia. Tuy nhiên, bên cạnh việc mở rộng, công tác giám sát, phân loại và kiểm soát phương tiện tại trạm thu phí cũng đặt ra những thách thức mới về chi phí vận hành, độ chính xác và khả năng xử lý thời gian thực.

1.1 Hình đường cao tốc


Hình 1.1

Hiện nay, một số trạm thu phí vẫn còn sử dụng các phương pháp kiểm soát truyền thống như kiểm tra thủ công, quét RFID đơn lẻ hoặc camera thông thường chưa tích hợp AI, dẫn đến các vấn đề như:

Xếp hàng kéo dài, gây ùn tắc cục bộ.

Sai sót trong phân loại phương tiện dẫn đến thu phí không chính xác.

Khó giám sát hành vi vi phạm hoặc xe cố tình tránh phí.

Do đó, nhu cầu áp dụng công nghệ thông minh, đặc biệt là thị giác máy tính và trí tuệ nhân tạo, vào hệ thống giám sát tại trạm thu phí đang trở nên cấp thiết.

1.2 Vấn đề nghiên cứu

Hệ thống giám sát tại trạm thu phí có thể trở nên hiệu quả hơn khi được tích hợp với camera thông minh và mô hình AI, đặc biệt là các mô hình học sâu (Deep Learning) có khả năng phân tích hình ảnh theo thời gian thực. Trong đó, mô hình YOLOv8 – phiên bản mới nhất trong chuỗi mô hình phát hiện đối tượng "You Only Look Once" – nổi bật với khả năng:

Nhận diện phương tiện (xe máy, ô tô con, xe tải, xe buýt...) với độ chính xác cao.

Xử lý thời gian thực, phù hợp cho các trạm thu phí nơi xe liên tục lưu thông.

Khả năng tích hợp vào các hệ thống camera giám sát sẵn có.

Hệ thống có thể phục vụ cho các mục tiêu sau:

Phân loại xe chính xác để tính phí đúng loại phương tiện.

Giám sát hành vi vi phạm như vượt trạm, xe không đủ điều kiện, hoặc chạy sai làn.

Lưu trữ dữ liệu cho việc phân tích lưu lượng, lên kế hoạch bảo trì, và hỗ trợ quản lý giao thông.

1.3 Đối tượng nghiên cứu

Mô hình YOLOv8 trong việc phát hiện và phân loại phương tiện giao thông.

Ứng dụng công nghệ thị giác máy tính (Computer Vision) và mạng nơ-ron tích chập (CNN).

Phân tích độ chính xác của hệ thống trong môi trường trạm thu phí thực tế.

1.4 Phạm vi nghiên cứu

Về mặt kỹ thuật: Tìm hiểu, đánh giá và thử nghiệm mô hình YOLOv8 trong phân loại các phương tiện qua hình ảnh từ camera trạm thu phí.

Về thực tiễn ứng dụng: Mô phỏng hoạt động của hệ thống tại trạm thu phí tại Việt Nam, so sánh với hệ thống các nước tiên tiến như Hàn Quốc, Nhật Bản.

Về hiệu suất hệ thống: Đo lường tốc độ xử lý, độ chính xác nhận diện trong điều kiện ánh sáng, thời tiết và mật độ xe khác nhau.

Về thách thức triển khai: Phân tích khó khăn về chi phí, tích hợp hạ tầng, bảo mật dữ liệu và xử lý thời gian thực.

1.5 Lý do chọn đề tài

Trong bối cảnh chuyển đổi số toàn diện trong ngành giao thông vận tải, các trạm thu phí truyền thống đang dần được thay thế bằng hệ thống thu phí không dừng (ETC). Tuy nhiên, để đảm bảo hiệu quả vận hành, các hệ thống giám sát cần được nâng cấp với khả năng tự động phát hiện và phân loại xe chính xác.

Phương pháp truyền thống như kiểm tra bằng mắt người, quét thẻ thủ công hay cảm biến đơn lẻ không còn đáp ứng được yêu cầu về tốc độ và độ tin cậy trong thời đại mới. Bên cạnh đó, các hành vi gian lận như tráo đổi biển số, xe đi sai làn, hoặc tránh trạm vẫn là vấn đề nhức nhối.

Việc ứng dụng mô hình YOLOv8 vào phân loại xe tại trạm thu phí là một giải pháp hiện đại, có khả năng nhận diện chính xác trong thời gian thực, phù hợp với điều kiện hạ tầng hiện nay và dễ dàng mở rộng quy mô.

Vì vậy, em lựa chọn đề tài:

“Ứng Dụng YOLOv8 Trong Phân Loại Xe Để Giám Sát Trong Trạm Thu Phí”

với các mục tiêu:

Tìm hiểu chuyên sâu về ứng dụng AI và thị giác máy tính trong giao thông.

Triển khai thử nghiệm mô hình trong tình huống mô phỏng thực tế tại trạm thu phí.

Đề xuất hướng cải tiến, phát triển hệ thống giám sát thông minh có tính ứng dụng cao tại Việt Nam.

Đây không chỉ là một đề tài mang tính học thuật mà còn có tiềm năng áp dụng thực tiễn cao, góp phần vào mục tiêu phát triển giao thông thông minh, an toàn và hiệu quả trong thời đại chuyển đổi số.

Chương 2: Cơ sở lý thuyết

2.1. Phát hiện đối tượng là gì?

Phát hiện đối tượng là một nhiệm vụ quan trọng trong thị giác máy tính, cho phép máy móc xác định và định vị các đối tượng cụ thể trong hình ảnh hoặc video.

Không giống như phân loại hình ảnh, chỉ xác định sự hiện diện của một đối tượng trong hình ảnh, phát hiện đối tượng vẽ các hộp giới hạn xung quanh mỗi đối tượng được phát hiện, chỉ định vị trí của nó.

Công nghệ này thu hẹp khoảng cách giữa cách máy móc nhận thức dữ liệu hình ảnh và cách con người hiểu môi trường xung quanh.

Về bản chất, phát hiện đối tượng kết hợp hai quy trình chính: phân loại và định vị.

Phân loại xác định những đối tượng nào có mặt (ví dụ: ô tô, người, cây), trong khi định vị xác định vị trí của những đối tượng này trong hình ảnh, thường bằng cách vẽ một hộp giới hạn xung quanh chúng.

Điều này thường đạt được bằng cách sử dụng các thuật toán phức tạp, thường dựa trên Mạng nơ-ron tích chập (CNN) , học cách nhận dạng các mẫu và đặc điểm đặc trưng cho các đối tượng khác nhau.

Độ chính xác của các mô hình phát hiện đối tượng thường được đánh giá bằng các số liệu như Giao điểm trên hợp (IoU) và Độ chính xác trung bình trung bình (mAP).



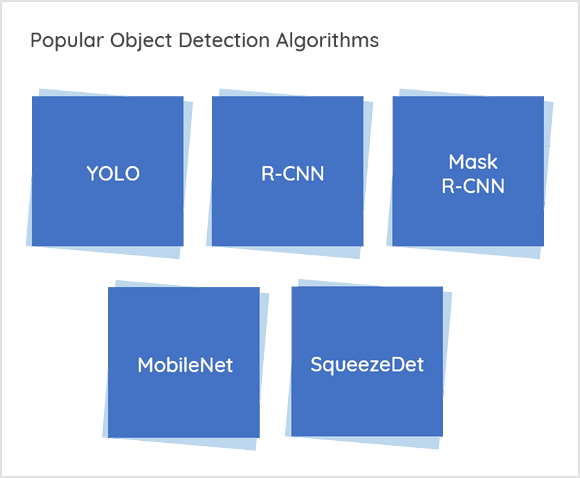
Hình 2.1

2.2 Các loại mô hình phát hiện đối tượng

Các mô hình phát hiện đối tượng có thể được phân loại thành hai loại chính:

Các mô hình sử dụng một giai đoạn.

Các mô hình sử dụng hai giai đoạn.



Hình 2.2

2.2.1 Các mô hình sử dụng một giai đoạn

**R-CNN (Region-based ConvNet)**

Đề xuất một số vùng tiềm năng bằng thuật toán thô khác, chẳng hạn selective search.

Dùng mạng CNN trích xuất đặc trưng từng vùng rồi phân loại bằng SVM.

**Fast-RCNN**

Đẩy tất cả các vùng (khoảng 2000) qua mạng trích xuất CNN cùng một lúc.

Crop thông tin ở lớp đầu ra của CNN thay vì crop vùng trên ảnh gốc như R-CNN.

Đẩy qua nhánh phân loại và nhánh hiệu chỉnh tọa độ box.

2.2.2 Các mô hình sử dụng hai giai đoạn

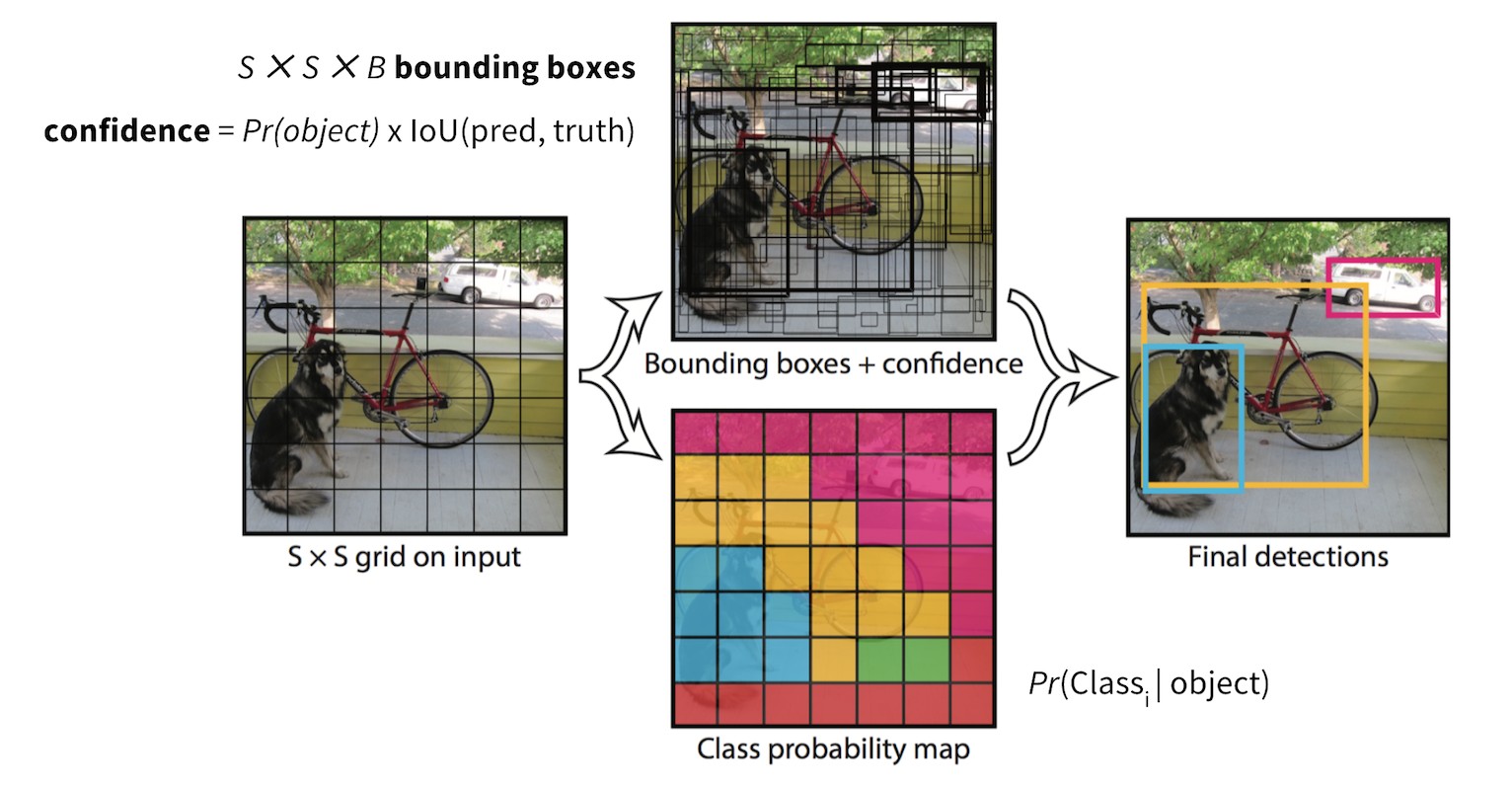
Ngược lại, mô hình một giai đoạn, như Ultralytics YOLO, SSD, cung cấp hiệu suất nhanh hơn bằng cách dự đoán trực tiếp các hộp giới hạn và xác suất lớp trong một lần chạy.

Các mô hình này thường đề xuất một lưới box dày đặc trên ảnh ban đầu, thường có bước nhảy đều (stride).

Từng box này sẽ được phân loại và hiệu chỉnh tọa độ (nếu box chứa đối tượng) bằng mạng CNN.

Các mô hình một giai đoạn thường nhanh hơn và đơn giản hơn các mạng hai giai đoạn.

**YOLO- You Only Look Once**



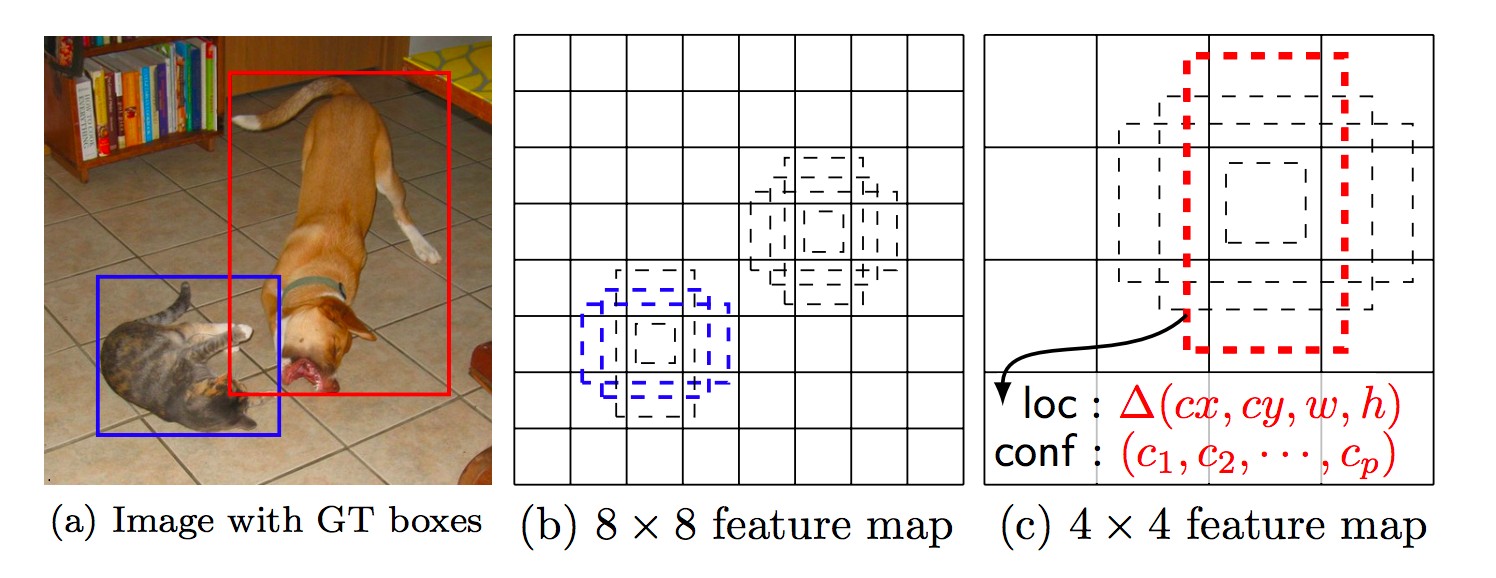
Hinh 2.3

**SSD: Single Shot Detector**

Tương tự YOLO nhưng lưới box dày đặc hơn, có nhiều lưới với các kích thước box khác nhau.

Kiến trúc mạng backbone khác với YOLO.

Data augmentation + Hard negative mining.



Hình 2.4

2.3 Mạng YOLO (You Only Look Once):

2.3.1 Giới thiệu về 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 (Conv) và connected layers. Trong đó các convolutional layers sẽ trích xuất ra các 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.

Mô hình YOLO được giới thiệu lần đầu tiên vào năm 2016 bởi Joseph Redmon và các cộng sự trong bài báo "You Only Look Once: Unified, Real-Time Object Detection".



Hình 2.5

2.3.2 Các phiên bản YOLO

**Bảng 2.1**

|  |  |  |
| --- | --- | --- |
| **Phiên bản** | **Năm phát hành** | **Điểm khác biệt** |
| YOLOv1 | 2016 | Phiên bản đầu tiên, tốc độ nhanh nhưng độ chính xác thấp |
| YOLOv2 | 2017 | Cải thiện tốc độ và độ chính xác so với YOLOv1, sử dụng Darknet-19 làm backbone |
| YOLOv3 | 2018 | Cải thiện hiệu suất và khả năng khái quát hóa so với YOLOv2, sử dụng Darknet-53 làm backbone |
| YOLOv4 | 2020 | Cải thiện độ chính xác và tốc độ so với YOLOv3, hỗ trợ nhiều mô hình backbone hơn |
| YOLOv5 | 2021 | Cải thiện tốc độ và hiệu quả so với YOLOv4, đơn giản hóa mã nguồn và dễ sử dụng hơn, sử dụng PyTorch |
| YOLOv6 | 2022 | Cải thiện hiệu suất và khả năng khái quát hóa so với YOLOv5, sử dụng mô hình Transformer và Swin Transformer |
| YOLOv7 | 2022 | Một trong những ưu điểm chính của YOLO v7 là tốc độ. Nó có thể xử lý hình ảnh với tốc độ 155 khung hình mỗi giây, nhanh hơn nhiều so với các thuật toán phát hiện đối tượng hiện đại khác. |
| YOLOv8 | 2023 | Tối ưu thời gian xử lý |

2.3.3 Ứng dụng YOLO

**Giám sát an ninh:**

YOLO được sử dụng để phát hiện các đối tượng nghi ngờ trong các hệ thống giám sát an ninh, chẳng hạn như camera giám sát.

Ví dụ: YOLO có thể được sử dụng để phát hiện người xâm nhập, hành vi trộm cắp hoặc các hoạt động bất thường khác.

**Nhận diện khuôn mặt:**

YOLO được sử dụng để phát hiện khuôn mặt trong ảnh và video, giúp cho việc nhận diện khuôn mặt trở nên hiệu quả hơn.

YOLO có thể được sử dụng trong các hệ thống kiểm soát ra vào, thanh toán di động hoặc các ứng dụng xác thực danh tính khác.

**Phân loại ảnh:**

YOLO được sử dụng để phân loại ảnh tự động, giúp xác định các đối tượng trong ảnh một cách nhanh chóng và chính xác.

Ví dụ: YOLO có thể được sử dụng để phân loại ảnh sản phẩm, ảnh y tế hoặc ảnh phong cảnh.

**Xe tự lái:**

YOLO được sử dụng để phát hiện các đối tượng khác trên đường, chẳng hạn như xe cộ, người đi bộ và biển báo giao thông, giúp xe tự lái di chuyển an toàn hơn.

YOLO có thể được tích hợp vào hệ thống cảm biến của xe tự lái để nhận biết môi trường xung quanh và đưa ra quyết định lái xe phù hợp.

**Robot:**

YOLO được sử dụng để giúp robot nhận biết môi trường xung quanh và thực hiện các nhiệm vụ một cách hiệu quả hơn.

Ví dụ: YOLO có thể được sử dụng để giúp robot chọn và sắp xếp các đối tượng, hoặc để điều hướng robot trong môi trường phức tạp.

**Kiểm tra chất lượng sản phẩm:**

YOLO có thể được sử dụng để phát hiện các lỗi sản phẩm trong quá trình sản xuất.

**Giám sát y tế:**

YOLO có thể được sử dụng để phát hiện các dấu hiệu bất thường trong hình ảnh y tế.

**Nâng cao trải nghiệm người dùng:**

YOLO có thể được sử dụng để tạo ra các ứng dụng tương tác với người dùng dựa trên cử chỉ hoặc hành động.

**Ưu điểm của YOLO:**

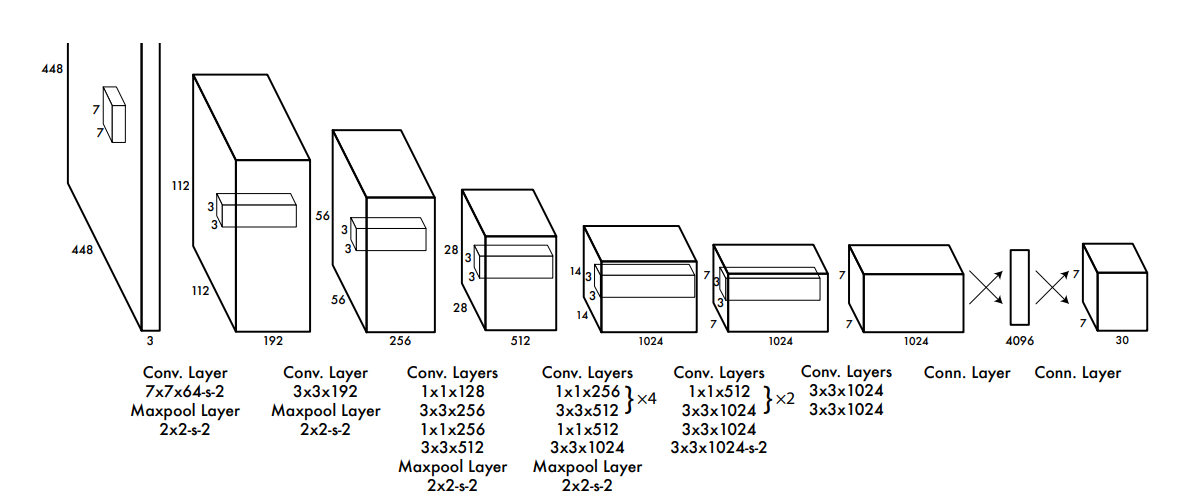
Tốc độ xử lý nhanh

Độ chính xác cao

Khả năng khái quát hóa tốt

Dễ dàng sử dụng.

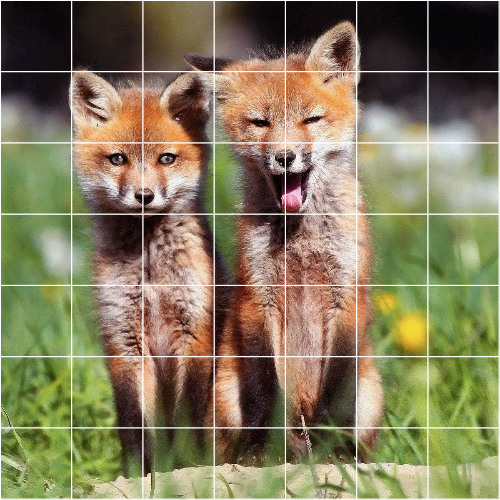
2.3.4 YOLOv1



Hình 2.6

24 lớp tích chập và 2 lớp FCL với input image 448\* 448, ta có được output là một vector với 4096 tham số được resize về 7\* 7\* 30

**Cách thức hoạt động**



Hình 2.7

Ý tưởng chia ảnh thành các grid cell để học vị trí của vật thể, nếu tâm của vật thể nằm tại ô nào thì ô đó sẽ đảm nhiệm việc detect ra bounding box cho vật thể ấy.

Mỗi grid cell sẽ dự đoán 2 bounding boxes và confidence score cho mỗi box đó.

Tổng bounding boxes cần dự đoán: 7x7x2 = 98 bounding boxes/ ảnh.

**Bounding boxes**

Bounding box là các hình chữ nhật bao xung quanh vật thể mang những thông tin về tọa độ và chiều dài chiều rộng của bounding box. Một bouding box sẽ có 5 giá trị thành viên, cụ thể:

Δx, Δy : Độ lệch giữa tọa độ tâm của bounding box dự đoán và tọa độ điểm bên trái trên cùng của grid cell mà nó thuộc về, giá trị của Δx,Δy vì vậy mà nằm trong khoảng [0,1].

Δw, Δh : Chiều rộng, chiều dài của bbox được chuẩn hóa theo kích thước của toàn bộ bức ảnh, giá trị của Δw, Δh nhờ vậy cũng bị ràng buộc trong khoảng [0,1]

c : Xác suất bên trong bouding box có vật thể là bao nhiêu.

Lý do độ lệch Δx, Δy được tính theo grid cell còn Δw, Δh lại được chuẩn hóa độ lớn của cả ảnh?

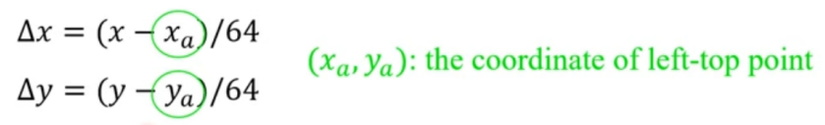
Vật thể sẽ được xác định vị trí theo grid cell, nghĩa là mô hình chỉ quan tâm việc dự đoán giá trị độ lệch ấy mà không cần quan tâm đến chính xác vị trí tâm vật thể ở đâu (tọa độ cụ thể).

Vị trí tâm của vật thể luôn có ràng buộc đến grid cell mà nó thuộc về, nhưng điều này cũng là một yếu điểm lớn của YOLOv1 vì khi một grid cell xác định một tâm vật thể và chỉ dự đoán xác suất một lần duy nhất dù cho có bao nhiêu bounding box trong grid cell => một grid cell tối đa chỉ dự đoán được một vật thể, cả ảnh tối đa chỉ dự đoán được 7\* 7 = 49 vật thể.

w, h được chuẩn hóa theo độ lớn của toàn ảnh vì khi đó giá trị của w, h nhỏ và việc dự đoán chiều rộng và chiều dài sẽ không bị ảnh hưởng bởi độ phân giải của ảnh.

Ví dụ ảnh đầu vào có là bao nhiêu thì model cũng chỉ việc dự đoán độ lệch chiều dài, chiều rộng trong khoảng [0,1] rồi về sau scale lại. Ngoài ra giá trị w,h trong khoảng [0,1] còn khiến việc tính loss và cập nhật tham số dễ dàng, nhanh chóng hơn.

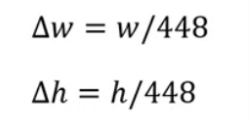
Công thức chuẩn hóa x, y center về [0,1]:



Hình 2.8

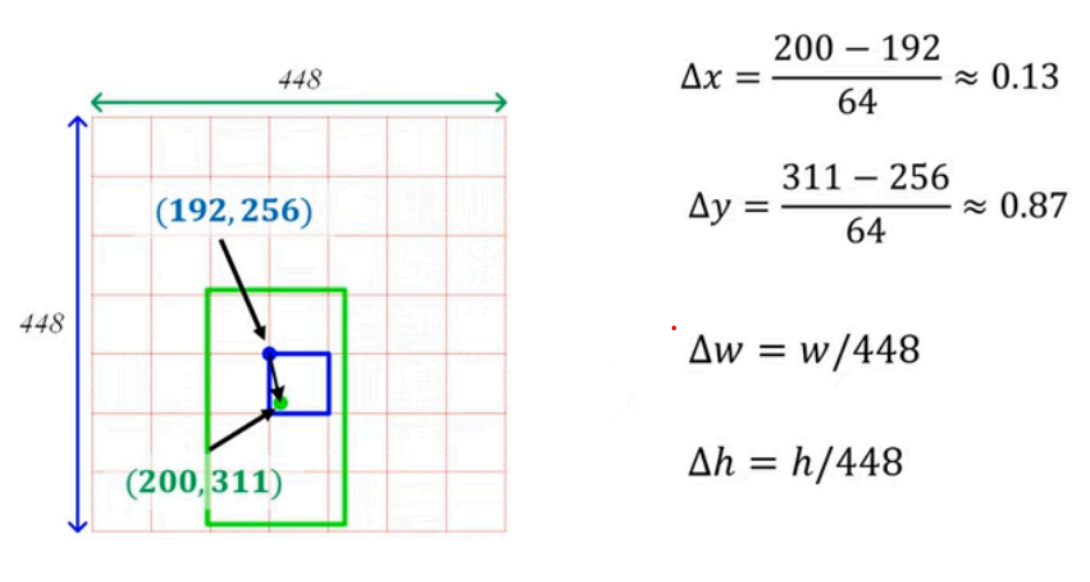
Tọa độ tâm của bounding box (x) trừ đi tọa độ của điểm góc trên cùng bên trái của grid cell, sau đó chia cho chiều rộng của một grid cell (trong ví dụ này là 64) với kích thước toàn ảnh là 448 \* 448.

Chuẩn hóa chiều dài chiều rộng của Bounding boxes:



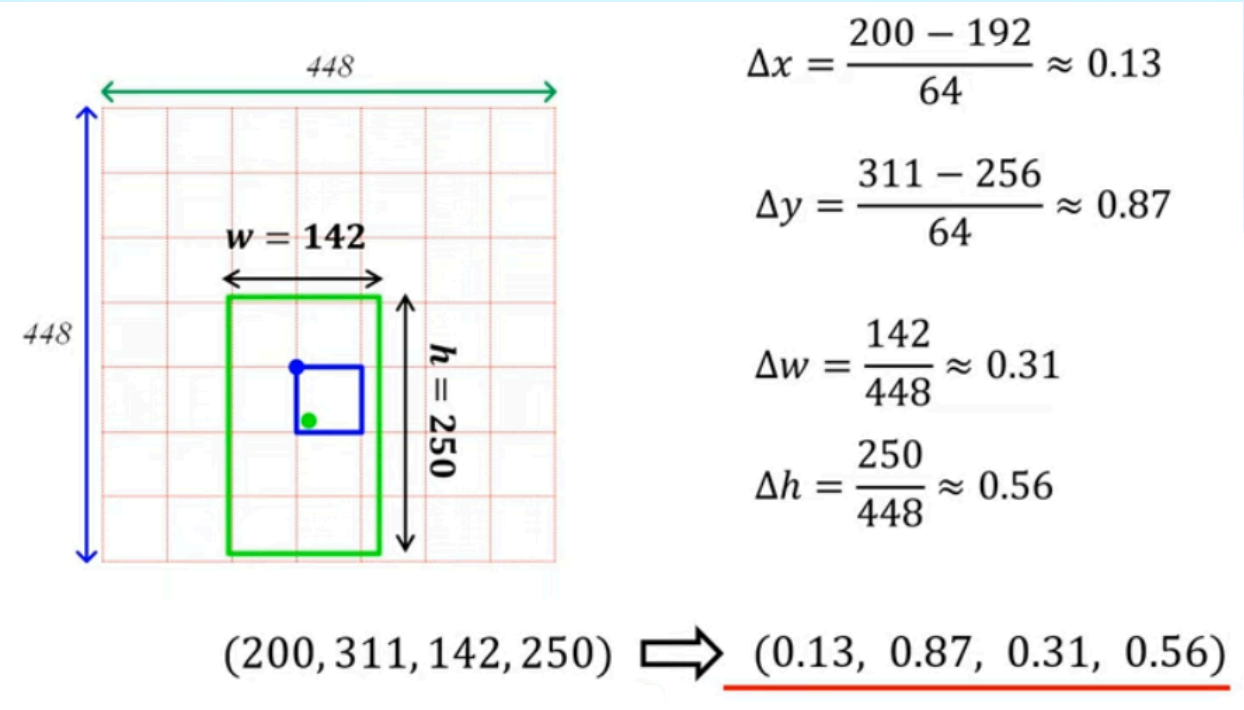
Hình 2.9

Ví dụ khi chuẩn hóa bằng công thức trên, với tọa độ tâm của vật thể là (200,311), Tọa độ góc trên cùng bên trái của grid cell chứa vật thể là (192,256), ta có 4 tọa độ của bounding box:



Hình 2.10

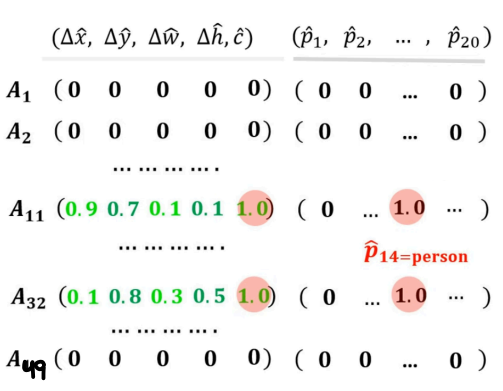
Ví dụ khi chuẩn hóa bằng công thức trên, với tọa độ tâm của vật thể là (200,311), Tọa độ góc trên cùng bên trái của grid cell chứa vật thể là (192,256), ta có 4 tọa độ của bounding box:



Hình 2.11

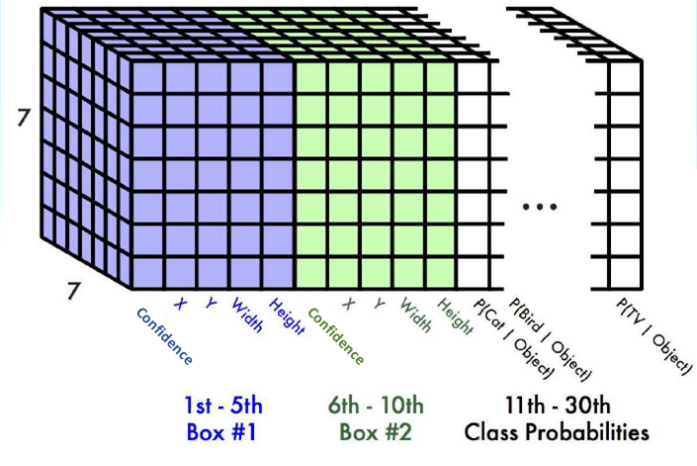
**Label encoding ground truth box**

Sau khi có các giá trị tọa độ bounding box thì tiến hành label encoding ground truth box chuẩn bị cho quá trình training.



Hình 2.12

**Predicted bounding box**

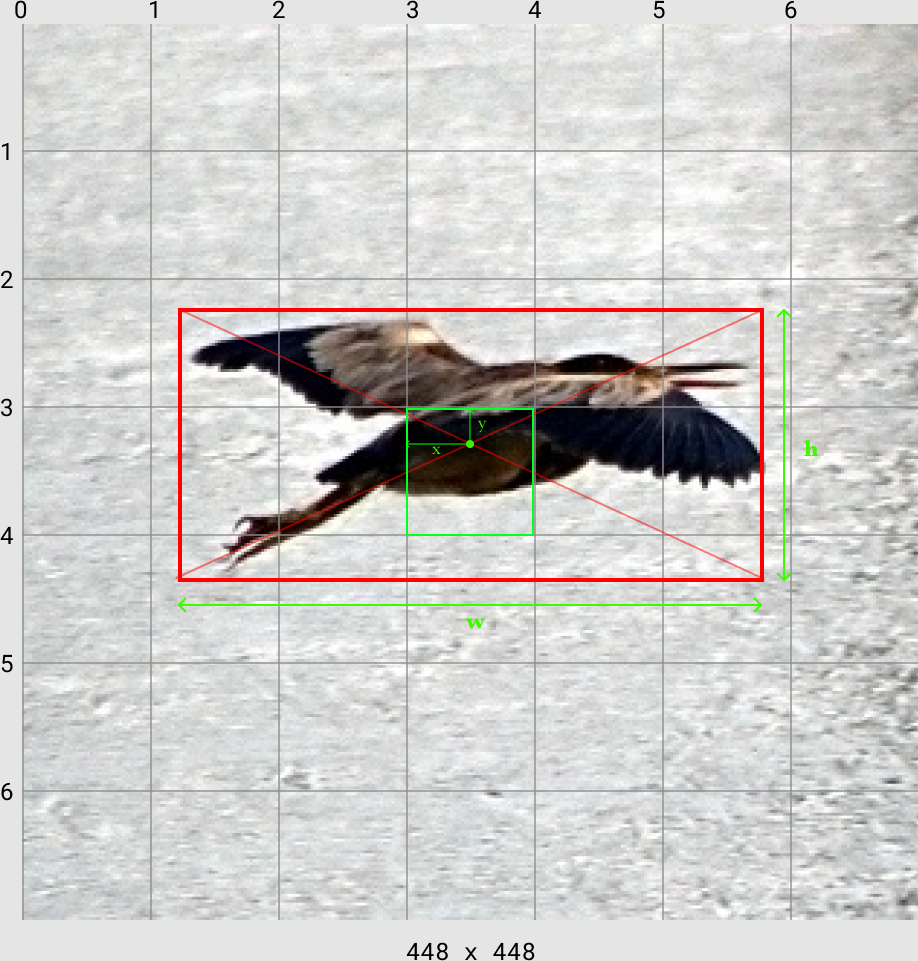


Hình 2.13

Thường một grid cell trong YOLOv1 sẽ dự đoán ra 2 bouding box, mỗi box có 5 giá trị ( x1, y1, w1, h1, c1), (x2, y2, w2, h2, c2).

Khác với labeled ground truth bbox, cho dù có 2 hay nhiều bounding box dự đoán trong một grid cell thì các nhãn label cũng chỉ được dự đoán một lần => tối đa một grid cell chỉ dự đoán được một object.

Như vậy, giả sử ta có 20 class trong bộ dữ liệu, cộng thêm 10 giá trị đến từ cả hai bounding box => Một grid cell mã hóa ra được một vector có độ lớn 20 + 10 = 30.



Hình 2.14

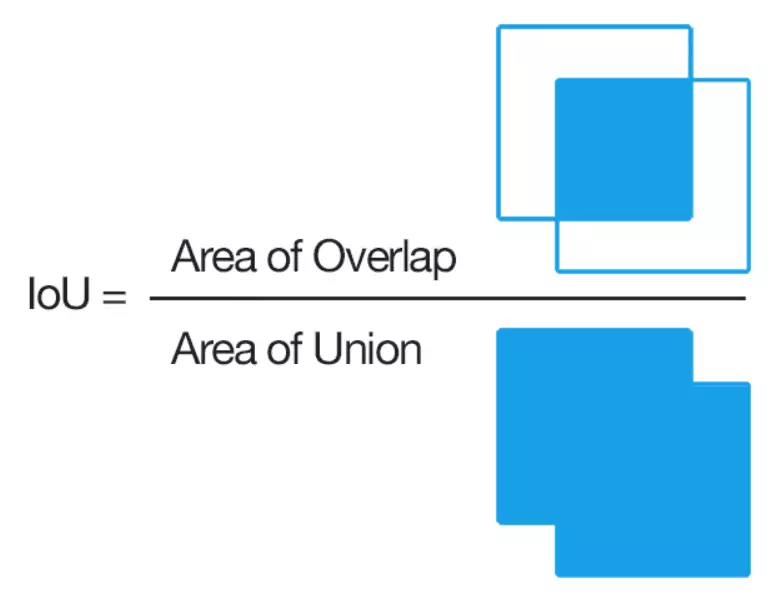
**Class confidence (Độ tự tin grid cell có chứa vật thể hay không)**



Hình 2.15

Pr(Object) là xác suất có vật thể ( 0 hoặc 1)

IoU(intersection over union) là công thức tính xem mức độ trùng nhau của bounding box dự đoán và ground truth box.



Hình 2.16

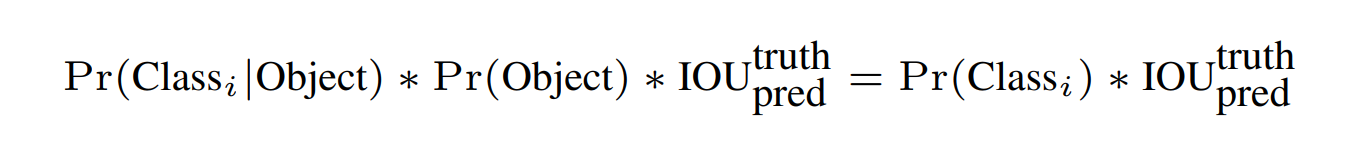
Phần tử là diện tích phần giao nhau của predicted bounding box và ground truth box, phần mẫu là tổng diện tích của 2 box trên hợp lại.

Người ta sẽ quy định một Ngưỡng, thường thì IoU score >= 0.5 có nghĩa là predicted bouding box có độ tương đồng cao với ground truth box.

Từ công thức tính trên ta thấy objectness confidence chính là IoU của box đó với bất kỳ ground truth box nào rồi từ đó chọn ra kết quả có IoU cao nhất để làm objectness confidence cho bbox.

Do YOLOv1 thường sẽ dự doạn 2 box cho một ô, nghĩa là không quan tâm ô đó có vật thể hay không thì model vẫn sẽ dự đoán ra 2 bouding box, khi ấy một ngưỡng IoU được người dùng đặt ra ( thường là 0.5), nếu ngưỡng IoU của predicted bounding box nào nhỏ hơn 0.5 thì sẽ bị loại bỏ.

**Confidence score (xác suất box đó chứa loại vật thể gì)**



Hình 2.17

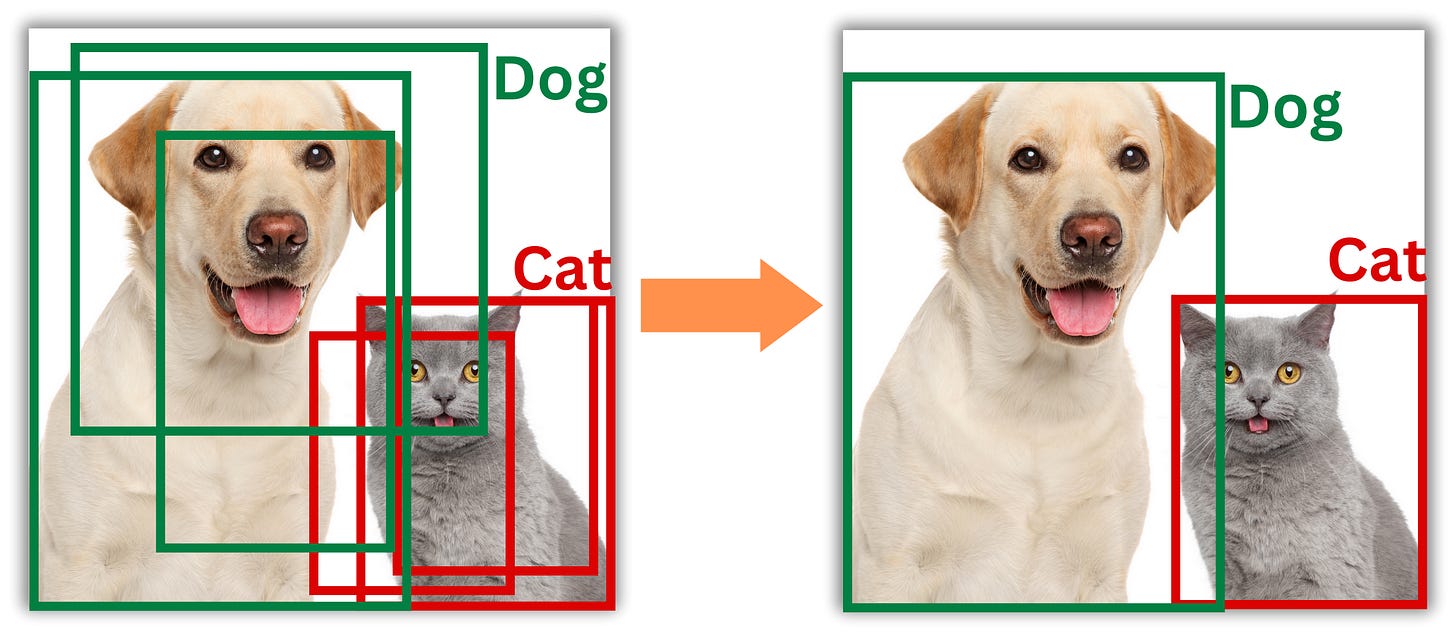
p(0,...,i,...n) được tính bằng xác suất có điều kiện giữa việc có vật thể thì nhãn của vật thể đó là gì, nhân với objectness confidence vừa tính ở trên. Khi có object, Pr(Object) = 1, Pr(Class(i)|Object) = Pr(Class(i)) nên ta có công thức tính xác suất của từng nhãn ở phía bên phải.

Confident score vừa thể hiện mức độ tương đồng của predicted bounding box và xác suất của nhãn được gán cho box là bao nhiêu.

**Non-max suppression**

Sau quá trình predicted bounding box của model, vẫn còn nhiều bounding box trùng lặp lên nhau với ngưỡng IoU với ground truth box cao.

Vậy làm thế nào để ta có thể loại bỏ đi những box dự đoán cùng một vật thể, chỉ giữ lại 1 box với độ chính xác cao nhất cho một vật thể. Câu trả lời là sử dụng thuật toán Non maximum Suppression.

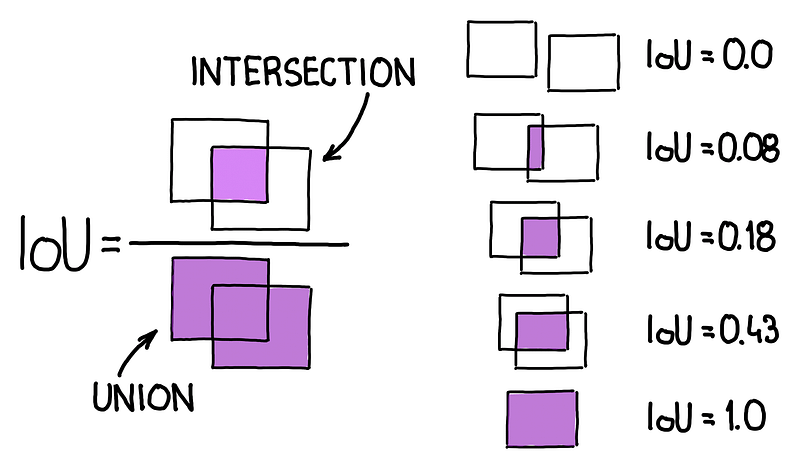


Hình 2.18

**Các bước:**

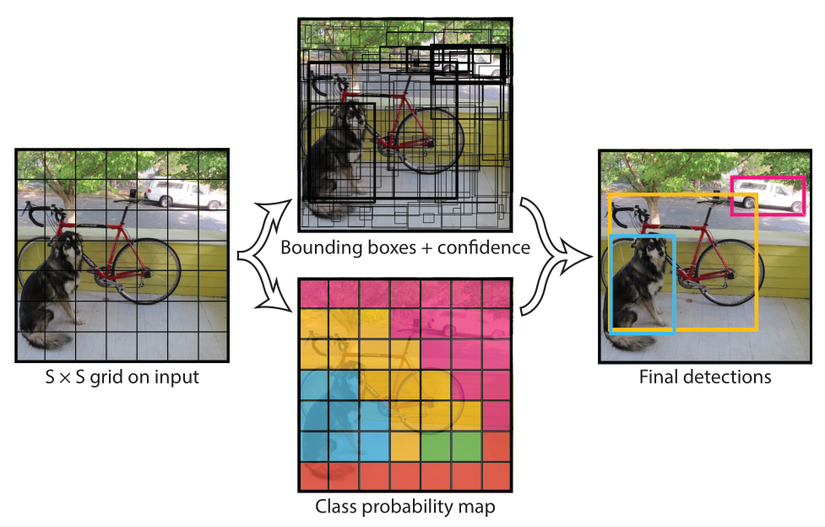
Bước 1: Ta có một list các bounding box còn lại gọi là P(box1, box2, box3,....) được sắp xếp theo thứ tự giảm dần về objeccness score. Những box đầu có độ tự tin cao sẽ được ưu tiên trước. Tiếp theo ta tạo một danh sách S gồm các box chuẩn đầu ra cuối cùng.

Bước 2: Lấy box đầu tiên từ tập P bỏ vào tập S, xóa box đó khỏi tập P sau đó tính tỷ lệ IoU với từng box còn lại trong tập P, nếu box nào trong tập P có tỷ lệ IoU so với box đang lấy làm chuẩn nhỏ hơn ngưỡng đặt ra (thường là 0,5) sẽ bị xóa khỏi tập P.



Hình 2.19

Bước 3: Nếu tập P vẫn còn phần tử thì tiếp tục lặp lại bước 2 cho đến khi P không còn phần tử nào nữa. Kết quả ta được tập S gồm các box phân biệt nhau cho từng vật thể.



Hình 2.20

2.3.5 YOLOv8

**Tổng quan về kiến trúc YOLOv8**

YOLOv8 là một mô hình anchor-free, nghĩa là nó không còn phụ thuộc vào các anchor box cố định như các phiên bản trước. Điều này giúp đơn giản hóa quá trình huấn luyện và cải thiện hiệu suất phát hiện.

YOLOv8 là một mô hình nhận dạng đối tượng dựa trên mạng convolutional neural network (CNN) được phát triển bởi Joseph Redmon và nhóm nghiên cứu của ông tại Đại học Washington.

YOLOv8 là phiên bản nâng cấp của YOLOv7, với khả năng nhận diện đối tượng nhanh hơn và chính xác hơn. Điều này được đạt được thông qua một số cải tiến, bao gồm mạng kim tự tháp đặc trưng, các mô-đun chú ý không gian và các kỹ thuật tăng cường dữ liệu tiên tiến.

Mô hình YOLOv8 sử dụng một mạng neural kiến trúc darknet-53 để trích xuất đặc trưng của hình ảnh và áp dụng thuật toán nhận dạng đối tượng YOLOv8 trên các đặc trưng đó.

**Key features:**

Tăng cường mô hình bằng cách thêm các kênh phân tán để tăng tốc độ tính toán.

Sử dụng kỹ thuật Attention để cải thiện khả năng nhận dạng đối tượng của mô hình.

Áp dụng phương pháp đào tạo mới để tăng tốc độ hội tụ.

Sử dụng kiến trúc mạng nơ-ron mới: Sử dụng kiến trúc YOLOv4 làm cơ sở để tăng hiệu suất và độ chính xác của mô hình.

Tích hợp cơ chế tự động điều chỉnh tỷ lệ tăng kích thước của hình ảnh đầu vào (AutoScale).

Hỗ trợ giám sát bằng video (Video Supervision): Mô hình có khả năng phát hiện và giám sát vật thể trong các video và đưa ra dự đoán liên tục trên toàn bộ video.

Tích hợp công nghệ Ensemble.

Tính năng điều chỉnh tỷ lệ tự động (AutoAnchor): Cải thiện việc phát hiện đối tượng với nhiều tỷ lệ khác nhau.

**Progression Timeline**



Hình 2.21

YOLOv8 có thể được chia thành 3 thành phần chính:

**A. Backbone (Xương sống – Trích xuất đặc trưng)**

Backbone trong YOLOv8 được xây dựng trên kiến trúc CSP (Cross Stage Partial) cải tiến, giúp:

Trích xuất đặc trưng hiệu quả hơn.

Giảm độ phức tạp mô hình (số tham số).

Tăng tốc độ suy luận.

Trong YOLOv8, backbone thường được gọi là C2f module, là phiên bản nâng cấp của C3 (từ YOLOv5). C2f tối ưu hóa tốc độ tính toán và độ chính xác.

**B. Neck (Truyền tải và tổng hợp đặc trưng)**

Phần Neck sử dụng FPN + PAN (Feature Pyramid Network + Path Aggregation Network) để:

Kết hợp đặc trưng từ nhiều tầng (đa cấp).

Phát hiện vật thể ở nhiều kích thước khác nhau (nhỏ – vừa – lớn).

Cải thiện khả năng định vị chính xác.

YOLOv8 vẫn giữ nguyên triết lý sử dụng Neck nhẹ và hiệu quả, nhấn mạnh khả năng suy luận thời gian thực.

**C. Head (Dự đoán kết quả)**

Phần Head chịu trách nhiệm:

Dự đoán hộp bao (bounding boxes).

Phân loại đối tượng.

Tính toán độ tin cậy (confidence score)

Khác với các phiên bản YOLO trước (sử dụng anchor-based), YOLOv8 sử dụng anchor-free detection head – giúp:

Đơn giản hóa quá trình training.

Giảm lỗi liên quan đến anchor.

Tăng hiệu suất trên các đối tượng mới/lạ.

**Mô hình đầu ra (Output)**

YOLOv8 dự đoán đầu ra theo dạng [x\_center, y\_center, width, height, confidence, class\_scores...] cho mỗi ô lưới, không còn dựa vào preset anchor.

Output kích thước: [batch, num\_predictions, 4 + 1 + num\_classes]

Trong đó:

4: tọa độ hộp (x, y, w, h)

1: độ tin cậy (objectness)

num\_classes: xác suất cho mỗi lớp

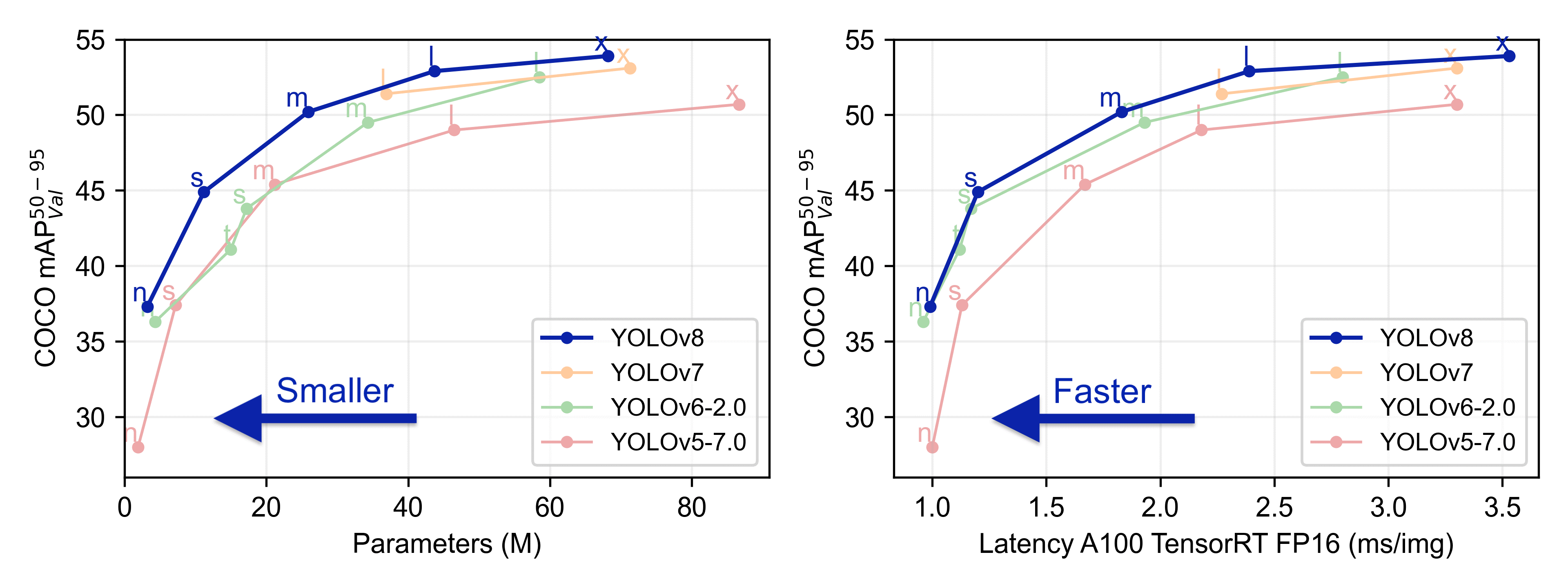
**Ưu điểm và hạn chế của YOLOv8**

**Ưu điểm**

Tốc độ: YOLOv8 được đánh giá là nhanh chóng và thời gian phản hồi thấp, giúp xử lý các tác vụ nhận diện đối tượng và phân-segment ảnh trong thời gian thực.

Độ chính xác: YOLOv8 được xây dựng trên các tiến bộ về học sâu và thị giác máy tính, đảm bảo độ chính xác cao trong việc nhận diện đối tượng.

Sự linh hoạt: YOLOv8 hỗ trợ việc nhận diện đối tượng và phân-segment trên cả GPU và CPU, tận dụng các công nghệ như TensorRT của Nvidia và OpenVino của Intel.



Hình 2.22

**Hạn chế**

Để sử dụng YOLOv8 hiệu quả cần phải:

Có kiến thức chuyên sâu về Machine Learning, Deep Learning và các thuật toán liên quan.

Cần phải được huấn luyện trên một tập dữ liệu đủ lớn và đa dạng để đạt được hiệu quả cao nhất.

Yêu cầu các tài nguyên tính toán cao để đạt được tốc độ xử lý nhanh và chính xác.

Thuật toán YOLOv8 không phải là mã nguồn mở và chỉ có sẵn thông qua các thỏa thuận cấp phép với người tạo ra nó, Joseph Redmon.

YOLOv8 có thể không hoạt động tốt trong tất cả các môi trường và có thể cần thêm điều chỉnh hoặc tối ưu hóa để đạt được hiệu suất tối ưu.

2.4 Tổng quan về bài toán phân loại xe trong giám sát trạm thu phí

2.4.1 Ứng dụng phát hiện đối tượng trong giám sát và phân loại xe

Trong hệ thống giao thông hiện đại, trạm thu phí không chỉ đóng vai trò thu ngân sách mà còn là nơi quản lý và điều phối luồng giao thông. Một trong những yêu cầu quan trọng tại các trạm thu phí là phân loại phương tiện một cách chính xác để áp dụng mức phí phù hợp, đồng thời hỗ trợ công tác giám sát, chống gian lận và nâng cao hiệu quả vận hành hệ thống thu phí tự động (ETC).

Mỗi loại xe (xe máy, ô tô con, xe tải nhẹ, xe tải nặng, container, xe buýt...) đều có mức phí và điều kiện lưu thông khác nhau, do đó việc phân loại sai có thể dẫn đến thất thoát doanh thu hoặc khiếu nại từ người dân. Trong thực tế, việc kiểm tra bằng mắt thường hoặc các cảm biến đơn giản dễ xảy ra nhầm lẫn, đặc biệt trong điều kiện lưu lượng cao hoặc xe bị thay đổi hình dạng (lắp mui, chở quá tải...).

Với sự phát triển của thị giác máy tính (Computer Vision), đặc biệt là các mô hình phát hiện đối tượng (Object Detection) như YOLOv8, việc tự động nhận diện và phân loại phương tiện tại trạm thu phí đã trở nên khả thi, chính xác và có thể hoạt động trong thời gian thực.

Quy trình ứng dụng mô hình phân loại xe tại trạm thu phí gồm các bước:

Ghi nhận video/ảnh từ camera trạm thu phí.

Phát hiện và phân loại phương tiện trong từng khung hình thông qua mô hình YOLOv8.

Gắn nhãn loại xe tương ứng với cơ sở dữ liệu thu phí.

Gửi thông tin đến hệ thống quản lý thu phí để áp dụng mức phí chính xác hoặc kiểm tra vi phạm (xe vượt trạm, không dán thẻ, v.v.).

Ứng dụng này không chỉ hỗ trợ tăng tốc độ xử lý tại trạm, mà còn giúp giảm thiểu sai sót trong phân loại, tăng cường khả năng phát hiện các hành vi bất thường hoặc gian lận thu phí, từ đó góp phần nâng cao hiệu quả và minh bạch trong quản lý giao thông.

2.4.2 Thách thức và định hướng nghiên cứu

**Thách thức:**

Việc ứng dụng mô hình phát hiện đối tượng như YOLOv8 trong giám sát trạm thu phí thực tế gặp phải một số thách thức lớn, bao gồm:

Môi trường vận hành phức tạp: Các phương tiện tại trạm có thể bị che khuất, chạy gần nhau hoặc thay đổi hình dạng (lắp mái, chở hàng...).

Đa dạng chủng loại phương tiện: Khó phân biệt giữa các loại xe tương tự như xe tải nhẹ và bán tải, hoặc SUV với xe khách nhỏ.

Điều kiện ánh sáng thay đổi: Hoạt động vào ban đêm hoặc trong điều kiện thời tiết xấu (mưa, sương mù) làm giảm chất lượng ảnh đầu vào.

Yêu cầu xử lý thời gian thực: Hệ thống cần phát hiện và phản hồi nhanh chóng để không gây cản trở lưu thông.

Thiếu bộ dữ liệu chuyên biệt: Phần lớn các mô hình quốc tế chưa được huấn luyện với hình ảnh phương tiện đặc thù tại Việt Nam.

**Định hướng nghiên cứu:**

Việc ứng dụng mô hình phát hiện đối tượng như YOLOv8 trong giám sát trạm thu phí thực tế gặp phải một số thách thức lớn, bao gồm:

Môi trường vận hành phức tạp: Các phương tiện tại trạm có thể bị che khuất, chạy gần nhau hoặc thay đổi hình dạng (lắp mái, chở hàng...).

Đa dạng chủng loại phương tiện: Khó phân biệt giữa các loại xe tương tự như xe tải nhẹ và bán tải, hoặc SUV với xe khách nhỏ.

Điều kiện ánh sáng thay đổi: Hoạt động vào ban đêm hoặc trong điều kiện thời tiết xấu (mưa, sương mù) làm giảm chất lượng ảnh đầu vào.

Yêu cầu xử lý thời gian thực: Hệ thống cần phát hiện và phản hồi nhanh chóng để không gây cản trở lưu thông.

Thiếu bộ dữ liệu chuyên biệt: Phần lớn các mô hình quốc tế chưa được huấn luyện với hình ảnh phương tiện đặc thù tại Việt Nam.

Chương 3: Cơ sở thực tiễn

3.1. Thực trạng hạ tầng giao thông tại Việt Nam

Trong bối cảnh đô thị hóa diễn ra nhanh chóng cùng với sự gia tăng mạnh mẽ của lưu lượng phương tiện, hệ thống hạ tầng giao thông tại Việt Nam đang chịu áp lực lớn. Đặc biệt, tại các tuyến đường cao tốc, quốc lộ và khu vực đô thị lớn, mặt đường thường xuyên xuất hiện các hiện tượng:

Lún, nứt, bong tróc do xe có tải trọng lớn lưu thông với tần suất cao.  
 Hư hỏng không được phát hiện và xử lý kịp thời, ảnh hưởng đến an toàn giao thông và hiệu quả lưu thông.  
 Công tác kiểm tra, giám sát chủ yếu mang tính thủ công, tốn kém nhân lực và thời gian.

Tình trạng này không chỉ ảnh hưởng đến người tham gia giao thông mà còn gây tổn thất kinh phí cho nhà nước, làm tăng nguy cơ tai nạn giao thông và chi phí bảo trì, nâng cấp đường bộ.

3.2. Thách thức trong công tác giám sát và bảo trì hiện nay

Phần lớn hoạt động bảo trì đường bộ hiện nay vẫn dựa vào các phương pháp thủ công:

Tuần tra trực tiếp hoặc chờ phản ánh từ người dân.  
 Ghi nhận hư hỏng bằng mắt thường hoặc thiết bị thô sơ, thiếu tính liên tục và độ chính xác.  
 Thiếu hệ thống thống kê cụ thể nguyên nhân gây hư hỏng; đặc biệt, chưa có cơ chế theo dõi và đánh giá mức độ tác động của từng loại phương tiện.

Do đó, việc áp dụng công nghệ vào công tác giám sát hạ tầng là xu hướng tất yếu, giúp nâng cao hiệu quả, độ chính xác và giảm chi phí vận hành.

3.3. Vai trò của công nghệ Deep Learning trong giám sát giao thông

Trong những năm gần đây, sự phát triển mạnh mẽ của trí tuệ nhân tạo (AI), đặc biệt là Deep Learning, đã mở ra nhiều cơ hội mới trong lĩnh vực giám sát và bảo trì hạ tầng giao thông. Các mô hình YOLO (You Only Look Once) cho phép phát hiện đối tượng trong thời gian thực, phù hợp với các nhiệm vụ sau:

Nhận dạng và phân loại phương tiện giao thông từ video thu được qua camera giám sát.  
 Thống kê tần suất xuất hiện và loại phương tiện gây hư hỏng mặt đường.  
 Ước lượng mức độ hư hại tích lũy theo thời gian, đưa ra cảnh báo bảo trì khi vượt ngưỡng cho phép.

Đây là nền tảng cho các hệ thống giám sát thông minh có khả năng hoạt động tự động và liên tục.

3.4. Giải pháp đề xuất từ đề tài

Dựa trên các hạn chế hiện tại và tiềm năng ứng dụng của Deep Learning, nhóm thực hiện đề xuất một hệ thống sử dụng mô hình YOLOv8 nhằm:

Nhận dạng các phương tiện giao thông (car, truck, bus, motorcycle) từ video thực tế.  
 Tính toán điểm hư hại dựa trên loại phương tiện và vùng ảnh hưởng trên khung hình.  
 Tổng hợp điểm hư hại tích lũy và phát cảnh báo bảo trì nếu vượt ngưỡng.

3.4.1. Mô hình tổng quát

Hệ thống đề xuất bao gồm ba giai đoạn chính:

Tiền xử lý video đầu vào: Chuyển đổi video thành các khung hình (frame), chọn lọc các frame cách đều nhằm giảm tải tính toán.

Nhận dạng đối tượng bằng YOLOv8: Sử dụng mô hình YOLOv8 đã được huấn luyện để nhận diện các phương tiện giao thông (xe hơi, xe tải, container, v.v.)

Tính toán điểm hư hại và thống kê: Gán trọng số hư hại tương ứng cho từng loại phương tiện; đồng thời áp dụng kỹ thuật tracking để tránh đếm trùng phương tiện xuất hiện trong nhiều khung hình.

3.4.2. Công cụ và thư viện sử dụng

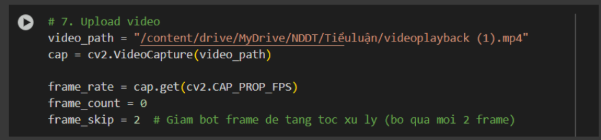
Ngôn ngữ: Python.

Thư viện chính:  
 Ultralytics: sử dụng mô hình YOLOv8.  
 Opencv-python: xử lý video và khung hình.  
 Matplotlib: trực quan hóa kết quả.  
 Collections, uuid, numpy: hỗ trợ quản lý ID phương tiện và tính toán điểm hư hại.

3.4.3. Tiền xử lý video đầu vào

Video đầu vào được chia thành các khung hình với khoảng cách đều nhau, nhằm:

Tiết kiệm tài nguyên xử lý.  
 Tránh trùng lặp quá mức giữa các frame gần nhau.



Hình 3.1

3.4.4. Nhận dạng đối tượng với YOLOv8

Sử dụng mô hình YOLOv8n để nhận diện các phương tiện giao thông trên từng khung hình. Mô hình sẽ xuất ra các thông tin sau:

Loại phương tiện: car, bus, truck  
 Tọa độ bounding box: dùng để xác định vị trí của phương tiện trong khung hình  
 Độ tin cậy (confidence): chỉ giữ lại các kết quả có độ tin cậy lớn hơn 0.5



Hình 3.2

3.4.5. Tính toán điểm hư hại, tránh để trùng

Để tính điểm hư hại, mỗi loại phương tiện sẽ được gán một trọng số như sau:

Xe hơi (car): 2.5 điểm  
 Xe buýt (bus): 4.8 điểm  
 Xe tải (truck): 5.0 điểm

Hệ thống sử dụng kỹ thuật tracking để theo dõi phương tiện qua nhiều khung hình, nhằm tránh việc đếm trùng khi cùng một phương tiện xuất hiện liên tục.

Áp dụng kỹ thuật object tracking đơn giản bằng ID tạm thời giúp tránh đếm trùng xe khi xuất hiện trong nhiều frame gần nhau. Mỗi phương tiện được gán một ID và so sánh bounding box giữa các frame để xác định phương tiện đã tồn hay mới.   
3.4.6. Ngưỡng cảnh báo và kết quả  
 Tổng điểm gây hư hỏng/ hư hại được cộng dồn theo từng đoạn video. Khi  
vượt ngưỡng quy định, hệ thống sẽ:  
 Ghi nhận sự kiện vượt ngưỡng  
 Lưu frame gây hư hại nghiêm trọng  
 Gửi cảnh báo  
 Kết quả cuối cùng bao gồm:  
 Biểu đồ thể hiện số lượng từng loại xe gây hư hỏng cho đoạn video  
đầu vào



Hình 3.3

Chương 4: Thực nghiệm

4.1 Sử dụng YOLOv8 trong bài toán phân loại xe để lập kế hoạch bảo trì mặt đường

4.1.1 Lý do lựa chọn mô hình YOLOv8

YOLOv8 (You Only Look Once version 8) là phiên bản mới nhất trong họ mô hình YOLO – một trong những kiến trúc hàng đầu trong lĩnh vực nhận dạng đối tượng theo thời gian thực. So với các phiên bản trước như YOLOv5 hoặc YOLOv7, YOLOv8 được cải tiến cả về kiến trúc mạng lẫn tốc độ xử lý, mang lại độ chính xác cao hơn, khả năng tổng quát tốt hơn và tốc độ suy luận nhanh hơn. Bên cạnh đó, YOLOv8 được phát triển bởi Ultralytics – một nhóm nghiên cứu tích cực với mã nguồn mở, hỗ trợ đầy đủ các công cụ huấn luyện, kiểm thử và triển khai.

Với đặc điểm của bài toán cần xử lý dữ liệu ảnh thực tế có nhiều yếu tố gây nhiễu như ánh sáng, vật thể đa dạng và góc nhìn phức tạp, YOLOv8 là lựa chọn phù hợp nhờ:

Khả năng phát hiện đối tượng nhỏ và phức tạp tốt hơn.

Tốc độ nhanh, có thể triển khai trong hệ thống giám sát thời gian thực.

Hỗ trợ các chức năng phân loại, phát hiện (object detection), phân đoạn (segmentation) và theo dõi (tracking) trong cùng một nền tảng.

4.1.2 Ứng dụng trong bài toán phân loại xe để lập kế hoạch bảo trì mặt đường

Trong khuôn khổ tiểu luận này, YOLOv8 được sử dụng để nhận diện và giám sát tình trạng cơ sở hạ tầng đường bộ, cụ thể là phát hiện các hiện tượng bất thường như:

Vết nứt (cracks) trên mặt đường, bê tông hoặc vách hạ tầng.

Sụt lún (subsidence), bong tróc mặt đường.

Vật thể cản trở (obstacles) như cây đổ, đá rơi, xe bị tai nạn nằm trên đường.

Việc phát hiện kịp thời các đối tượng này cho phép cảnh báo sớm nguy cơ mất an toàn giao thông, từ đó hỗ trợ lực lượng chức năng can thiệp nhanh chóng, nâng cao hiệu quả giám sát và bảo trì.

4.1.3 Cách thức hoạt động của mô hình YOLOv8

Mô hình YOLOv8 hoạt động theo nguyên lý phân tích toàn bộ ảnh đầu vào chỉ trong một lần quét (single forward pass). Quá trình nhận dạng bao gồm các bước sau:

Tiền xử lý dữ liệu: Ảnh đầu vào từ camera giám sát được chuẩn hóa về kích thước phù hợp với mô hình (ví dụ: 640×640 pixels), sau đó chuyển đổi về dạng tensor để đưa vào mạng nơ-ron.

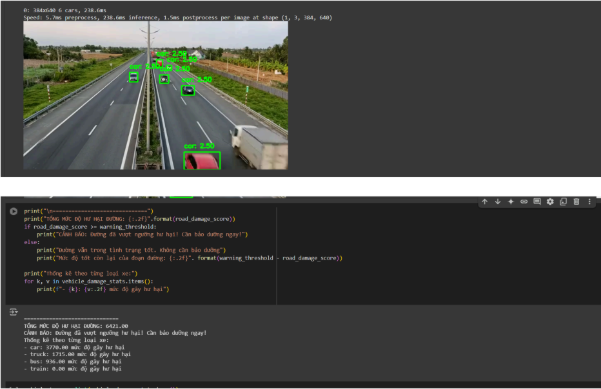
Dự đoán (inference): YOLOv8 chia ảnh thành nhiều lưới (grid cells) và dự đoán các bounding boxes kèm nhãn lớp và độ tin cậy (confidence score) tại từng ô lưới. Khác với các phiên bản trước, YOLOv8 sử dụng kiến trúc head mới, tăng khả năng nhận diện vật thể nhỏ.

Hậu xử lý: Áp dụng kỹ thuật Non-Maximum Suppression (NMS) để loại bỏ các hộp giới hạn (bounding boxes) trùng lặp, giữ lại các dự đoán có độ tin cậy cao nhất.

Hiển thị kết quả: Mô hình đầu ra là các hình ảnh hoặc video có gán nhãn các đối tượng phát hiện được (ví dụ: "crack", "obstacle", "pothole") với tọa độ và độ tin cậy tương ứng.

4.2 kết quả và thay đổi để tối ưu bài toán.

Trong quá trình thực hiện bài toán, ban đầu tụi em nghĩ cơ bản là chỉ cần đưa tạo ra đầu vào hệ số hư hại của xe và đặt ra ngưỡng khi các xe chạy qua thì sử dụng model để nhận dạng xe dựa vào hệ số hư hại ban đầu cung cấp mà mức độ gây hư hại đường do các loại xe tạo nên mà phát ra cảnh báo.



Hình 4.1

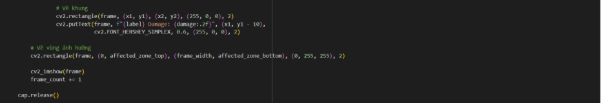
Tuy nhiên, chúng em nhận thấy các xe được tính trong từng frame bị lặp lại và kết quả bị cộng dồn lên khiến cho mức độ gây hư hại của xe đối với con đường vượt ngưỡng hơn so với thực tế rất nhiều .

4.2.1 Lần 1: Tạo vùng ảnh hưởng

Ở lần đầu thử nghiệm, chúng em nhận thấy đề tài bị mắc lỗi khá nặng so với thực tế ngoài đời sống. Nếu xe chạy trên một đoạn đường trong cùng một thời điểm, mà bị tính điểm gây hại lặp đi lặp lại rất nhiều là một điều gây nhiễu cho hệ thống quản lí cơ sở hạ tầng giao thông. Nên em đã tạo ra vùng ảnh hưởng bằng cách vẽ vùng ảnh hưởng ở giữa khung hình để tính các xe khi vào vùng ảnh hưởng sẽ được tính mức độ gây hại đối với con đường qua.



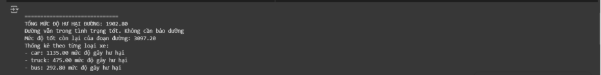
Hình 4.2



Hình 4.3



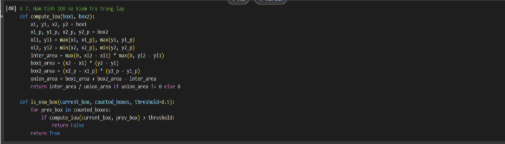
Hình 4.4



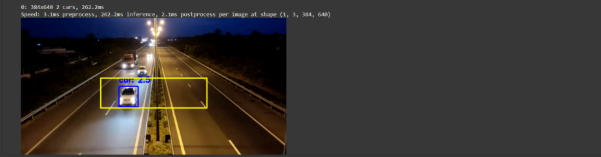
Hình 4.5

Ở lần kiểm tra này, bài toán có vẻ khả quan hơn một chút. Chỉ tính các xe mức độ gây hại của xe đối với con đường khi xe đó ở trong vùng ảnh hưởng màu vàng được vẽ ra qua từng frame. Tuy nhiên, mỗi frame được tính bằng 1s so với video và khi các xe còn ở trong vùng ảnh hưởng chưa vượt qua ở frame sau vẫn được cộng điểm gây hư hại mặc dù là được cộng ở frame trước đó, nên bài toán vẫn còn bị sai số một chút khi thống kê mức độ gây hại của từng loại xe đối với con đường.

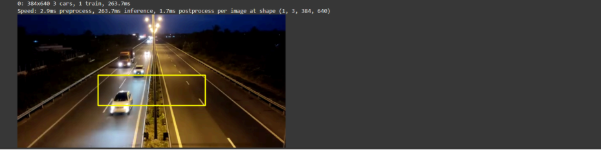
4.2.2 Lần 2: Dùng thuật toán IOU

Bằng việc cải thiện để tránh sự đếm trùng xe qua nhiều frame, chúng em đã sự dụng để thuật toán IOU (Intersection over Union) để so sánh giữa các bounding box giữa các frame.  
 Nếu xe nằm trong vùng ảnh hưởng thực hiện kiểm tra 2 bước như sau:  
 Kiểm tra xem có phải xe mới chưa đếm chưa ( IOU < 0.5)  
 Nếu là xe mới vào vùng => đếm, vẽ bounding box và text lên hình và ngược lại thì không tính.   


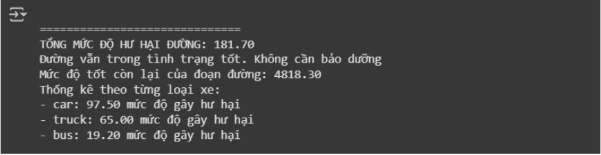
Hình 4.6



Hình 4.7



Hình 4.8



Hình 4.9

Kết quả đã khả quan và ổn định hơn sau các lần thực nghiệm và cải tiến giúp cho bài toán của chúng em có khả năng ứng dụng thực tế cao hơn so với các lần thực nghiệm đầu. Tối ưu được thời gian xử lí, bỏ đi các frame bị trùng lấp, lọc bớt đi các label bị trùng giúp cho model nhẹ hơn .

Chương 5: Kết luận

5.1 Nội dung đã thực hiện

Trong khuôn khổ tiểu luận, nhóm đã tiến hành nghiên cứu và triển khai bài toán bài toán phân loại xe để lập kế hoạch để bảo trì mặt đường bằng phương pháp nhận dạng đối tượng. Cụ thể, các nội dung chính đã thực hiện gồm:

Tìm hiểu cơ sở lý thuyết về nhận dạng đối tượng, đặc biệt là các mô hình nhận dạng đối tượng như YOLO.

Phân tích bài toán phân loại xe để lập kế hoạch để bảo trì mặt đường.

Lựa chọn và triển khai mô hình YOLOv8 cho bài toán nhận dạng, trải qua ba lần thử nghiệm với các cấu hình khác nhau.

Rút ra kết luận về hiệu quả của mô hình YOLOv8 trong việc phát hiện bất thường trên hình ảnh giao thông với độ chính xác và tốc độ cao.

5.2 Hạn chế

Bên cạnh những kết quả tích cực, tiểu luận vẫn còn một số hạn chế nhất định như sau:

Quy mô dữ liệu còn hạn chế, đặc biệt là số lượng ảnh về các loại hư hỏng ít gặp như sập lún nghiêm trọng hoặc vật thể lạ.

Điều kiện ảnh chưa đa dạng, chủ yếu là ảnh ban ngày, thiếu dữ liệu trong điều kiện ánh sáng yếu, mưa hoặc ban đêm.

Chưa triển khai mô hình trong môi trường thực tế, việc đánh giá mới chỉ thực hiện trên dữ liệu có sẵn, chưa tích hợp với hệ thống camera giao thông hoặc UAV ngoài thực địa.

Việc tinh chỉnh tham số và cải tiến mô hình còn ở mức cơ bản, chưa khai thác hết tiềm năng của YOLOv8 như phân đoạn ảnh (instance segmentation) hay theo dõi đối tượng (tracking).

5.3 Hướng phát triển

Trong các nghiên cứu tiếp theo hoặc nếu được triển khai thành một dự án thực tế, nhóm đề xuất một số hướng phát triển như sau:

Mở rộng tập dữ liệu bằng cách thu thập ảnh từ các nguồn đa dạng hơn, đặc biệt là ảnh trong điều kiện thời tiết và ánh sáng khác nhau.

Tích hợp hệ thống vào thiết bị thực tế, ví dụ như gắn camera lên drone để giám sát cầu đường, đường ray hoặc tuyến cao tốc.

Áp dụng các kỹ thuật tăng cường dữ liệu và học bán giám sát để nâng cao hiệu suất mô hình mà không cần tăng chi phí gán nhãn.

Kết hợp với mô hình theo dõi (object tracking) để theo dõi vật thể chuyển động hoặc bất thường lặp lại theo thời gian.

Tích hợp thêm mô-đun cảnh báo tự động, giúp hệ thống không chỉ nhận diện mà còn gửi cảnh báo ngay đến cơ quan chức năng hoặc hệ thống điều khiển trung tâm.

TÀI LIỆU THAM KHẢO

Ultralytics. (2023). *YOLOv8 - Ultralytics Documentation.* Retrieved from <https://docs.ultralytics.com>

Ultralytics. (2023). *YOLOv8 GitHub Repository.* GitHub. <https://github.com/ultralytics/ultralytics>