BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA** **CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN TỐT NGHIỆP**

**CHỦ ĐỀ**

**TÌM HIỂU YOLO, ỨNG DỤNG PHÁT TRIỂN HỆ THỐNG NHẬN DẠNG BẠO LỰC TẠI NƠI CÔNG CỘNG**

|  |  |
| --- | --- |
| Giảng viên hướng dẫn : | **TS. Đinh Đồng Lưỡng** |
| Sinh viên thực hiện: | **Nguyễn Quốc Thái** |
| MSSV: | **63131236** |

Khánh Hòa - 2025

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA** **CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN TỐT NGHIỆP**

**CHỦ ĐỀ**

**TÌM HIỂU YOLO, ỨNG DỤNG PHÁT TRIỂN HỆ THỐNG NHẬN DẠNG BẠO LỰC TẠI NƠI CÔNG CỘNG**

|  |  |
| --- | --- |
| Giảng viên hướng dẫn : | **TS. Đinh Đồng Lưỡng** |
| Sinh viên thực hiện: | **Nguyễn Quốc Thái** |
| MSSV: | **63131236** |

Khánh Hòa – Tháng 5/2025

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**Khoa/Viện**: Công nghệ thông tin

**PHIẾU THEO DÕI TIẾN ĐỘ VÀ ĐÁNH GIÁ ĐỒ ÁN / KHÓA LUẬN TỐT NGHIỆP  
*(Dùng cho CBHD và nộp cùng báo cáo ĐA/KLTN của sinh viên)***

Tên đề tài: Tìm hiểu yolo, ứng dụng phát triển hệ thống nhận dạng bạo lực tại nơi công cộng

Chuyên ngành: Công nghệ thông tin

Họ và tên sinh viên: Nguyễn Quốc Thái. Mã sinh viên: 63131236 Lớp: 63.CNTT-CLC

Người hướng dẫn (học hàm, học vị, họ và tên): TS. Đinh Đồng Lưỡng

Cơ quan công tác: Khoa Công nghệ thông tin - Trường Đại học Nha Trang

**Phần đánh giá và cho điểm của người hướng dẫn (tính theo thang điểm 10)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tiêu chí đánh giá** | **Trọng số**  **(%)** | **Mô tả mức chất lượng** | | | | **Điểm** |
| **Giỏi** | **Khá** | **Đạt yêu cầu** | **Không đạt** |
| **9 - 10** | **7 - 8** | **5 - 6** | **< 5** |
| Xây dựng đề cương nghiên cứu | 10 |  |  |  |  |  |
| Tinh thần và thái độ làm việc | 10 |  |  |  |  |  |
| Kiến thức và kỹ năng làm việc | 10 |  |  |  |  |  |
| Nội dung và kết quả đạt được | 40 |  |  |  |  |  |
| Kỹ năng viết và trình bày báo cáo | 30 |  |  |  |  |  |
| **ĐIỂM TỔNG** | | | | | |  |

*Ghi chú: Điểm tổng làm tròn đến 1 số lẻ.*

**Nhận xét chung** (sau khi sinh viên hoàn thành ĐA/KLTN):

…………………………………………………………………………………………….…………………………………………………………………………………………….…………………………………………………………………………………………………………….………………………………………………………………………………………………….Đồng ý cho sinh viên: Được bảo vệ:  Không được bảo vệ: 

*Khánh Hòa, ngày…….tháng…….năm………* **Cán bộ hướng dẫn** *(Ký và ghi rõ họ tên)*

**LỜI CAM ĐOAN**

Em xin cam kết rằng nghiên cứu "**Nghiên cứu học sâu, xây dựng công cụ nhận dạng chữ viết tay ứng dụng trong nhận dạng điểm số**" là thành quả quá trình của sự nỗ lực và phấn đấu của cá nhân em, Nguyễn Ngọc Hoài Sang, dưới sự hướng dẫn tận tình của Thầy Đinh Đồng Lưỡng. Tôi cam đoan rằng mọi kết quả trình bày trong nghiên cứu này là chân thực và không sao chép hay lấy từ bất kỳ nguồn tài liệu nào khác, tuy nhiên một phần dữ liệu dùng trong việc huấn luyện được giáo viên Hướng dẫn gửi cho sử dụng từ sản phẩm khoa học của Sinh viên Cao Minh Tiến, K61.

Tôi khẳng định rằng mình đã hoàn thành nghiên cứu này một cách trung thực, tuân thủ mọi quy định và yêu cầu của khoa Công nghệ thông tin và trường Đại học Nha Trang. Tất cả các tài liệu tham khảo được sử dụng trong nghiên cứu đã được trích dẫn đầy đủ, tuân thủ các quy tắc và quy định về trích dẫn tài liệu.

Trong trường hợp phát hiện bất kỳ sự sao chép nội dung hay kết quả nghiên cứu từ bất kỳ nguồn nào khác, tôi xin chịu hoàn toàn trách nhiệm và sẽ chấp nhận mọi hình thức kỷ luật từ Khoa và Nhà trường theo quy định hiện hành.

Tôi xin gửi lời cảm ơn chân thành đến Thầy Đinh Đồng Lưỡng vì sự hướng dẫn và hỗ trợ trong suốt quá trình thực hiện nghiên cứu này.

*Khánh Hòa, ngày 30 tháng 5 năm 2024*

**Sinh viên thực hiện**

Nguyễn Ngọc Hoài Sang

**LỜI CẢM ƠN**

Trước hết, em xin bày tỏ lòng biết ơn sâu sắc đến các thầy cô trong Khoa Công nghệ thông tin, trường Đại học Nha Trang, vì đã tạo điều kiện thuận lợi và hỗ trợ em trong suốt quá trình học tập.

Em đặc biệt cảm ơn anh Cao Minh Tiến đã nhiệt tình hỗ trợ và giúp đỡ em trong quá trình thực hiện đề tài.

Em cũng xin gửi lời cảm ơn chân thành đến Thầy Đinh Đồng Lưỡng, người đã tận tình hướng dẫn, cung cấp tài liệu và tạo điều kiện tốt nhất để em có thể phát triển công cụ thử nghiệm hỗ trợ nhập điểm và đối sánh dữ liệu từ ảnh bảng điểm viết tay. Sự dẫn dắt và động viên của Thầy đã giúp em hoàn thiện đề tài một cách chuyên nghiệp và đạt kết quả tốt nhất.

Cuối cùng, em xin chân thành cảm ơn Ban giám hiệu, Khoa Công nghệ thông tin và toàn thể Trường Đại học Nha Trang đã cung cấp môi trường học tập và nghiên cứu thuận lợi, tạo điều kiện tốt nhất cho em trong quá trình thực hiện đề tài này.

Trong quá trình thực hiện đề tài, do kiến thức và kinh nghiệm của em còn hạn chế, không thể tránh khỏi những thiếu sót. Vì vậy, em rất mong nhận được những góp ý quý báu từ các thầy cô để có thể hoàn thiện kiến thức và kỹ năng của mình hơn nữa.

Em xin chân thành cảm ơn!

Chương 1. TỔNG QUAN VỀ VẤN ĐỀ NGHIÊN CỨU

## 1.1 Bối Cảnh Và Ý Nghĩa Của Vấn Đề

Những năm gần đây, tình trạng bạo lực ở các nơi công cộng như đường phố, công viên hay trường học ngày càng tăng lên, khiến nhiều người lo lắng về an ninh và trật tự xã hội. Bạo lực không chỉ làm ảnh hưởng đến sức khỏe, tính mạng của mọi người mà còn tạo ra cảm giác bất an trong cộng đồng. Để giải quyết vấn đề này, việc giám sát và phát hiện sớm các hành vi bạo lực là điều rất cần thiết. Nhưng mà, cách giám sát truyền thống bằng camera thường cần người theo dõi liên tục, nên đôi khi phản ứng chậm hoặc bỏ sót những tình huống quan trọng vì lượng video quá nhiều.

May mắn là công nghệ trí tuệ nhân tạo (AI), đặc biệt là học sâu (Deep Learning) và thị giác máy tính (Computer Vision), đã phát triển mạnh mẽ. Nhờ vậy, việc tự động phát hiện bạo lực từ camera giám sát trở thành một giải pháp đầy hứa hẹn. Công nghệ này giúp giảm tải cho lực lượng an ninh, đồng thời tăng hiệu quả giám sát, bảo vệ cộng đồng nhanh hơn và chính xác hơn.

Đề tài của em tập trung vào việc xây dựng một hệ thống tự động phát hiện hình ảnh bạo lực ở nơi công cộng qua camera giám sát, dùng mô hình học sâu YOLOv11. Đây là một hướng đi thực tế và có ý nghĩa lớn, vừa ứng dụng được công nghệ hiện đại, vừa giải quyết vấn đề xã hội đang nóng hổi.

## 1.2 Động Lực Và Sự Phù Hợp Của Đề Tài

Lý do chọn đề tài này vì hai lý do chính: một là từ bản thân, hai là từ nhu cầu xã hội.

* **Động lực cá nhân**: Là sinh viên Công nghệ Thông tin, em rất thích áp dụng những gì đã học vào thực tế, nhất là trong lĩnh vực AI và xử lý ảnh. Đề tài này cho em cơ hội tìm hiểu sâu hơn về học sâu, đồng thời làm ra một thứ gì đó có ích cho mọi người.
* **Động lực xã hội**: Bạo lực đang tăng lên, nhu cầu đảm bảo an ninh cũng cao hơn bao giờ hết. Trong khi đó, công nghệ học sâu, đặc biệt là YOLOv11, lại đang rất phát triển và phù hợp để giải quyết bài toán này. Ứng dụng công nghệ vào thực tế như vậy là vừa kịp thời, vừa đúng xu hướng.

Đề tài không chỉ thiết thực mà còn kết hợp được công nghệ tiên tiến với nhu cầu xã hội, tạo ra một giải pháp có thể dùng được trong đời sống để tăng cường an ninh ở nơi công cộng.

**1.3 Mục Đích Và Kỳ Vọng Nghiên Cứu**

Mục đích chính của em là xây dựng một hệ thống tự động phát hiện hành vi bạo lực từ video camera giám sát ở nơi công cộng, dùng YOLOv11 kết hợp với mấy kỹ thuật xử lý ảnh và học sâu.

Cụ thể thì mình nhắm đến mấy mục tiêu sau:

* Nhận diện và phân loại các đối tượng liên quan đến bạo lực trong video.
* Xác định hành vi bạo lực dựa trên dữ liệu từ camera.
* Đánh giá xem hệ thống này có dùng được thực tế để hỗ trợ giám sát an ninh không.

Với kỳ vọng hệ thống sẽ hoạt động hiệu quả, dễ triển khai và có thể mở rộng sau này. Không chỉ phát hiện bạo lực, nó còn có thể được chỉnh sửa để phát hiện các hành vi bất thường khác, góp phần làm cho cộng đồng an toàn hơn.

## 1.4 Giới Hạn Và Đối Tượng Nghiên Cứu

Nghiên cứu của mình tập trung vào mấy thứ cụ thể sau:

* Đối tượng nghiên cứu: Video từ camera giám sát ở nơi công cộng như đường phố, công viên, trường học.
* Giới hạn:
  + Chỉ xử lý các hành vi bạo lực rõ ràng, không làm mấy tình huống phức tạp như bạo lực trong đám đông lớn, vì máy móc và dữ liệu của mình có hạn.
  + Dùng video có sẵn và video thực tế từ camera để thử nghiệm.

Giới hạn vậy để mình tập trung vào những tình huống cụ thể, đảm bảo hệ thống khả thi và hiệu quả trong điều kiện thực tế.

## 1.5 Cơ Sở Khoa Học Và Thực Tiễn

1.5.1 Các Nghiên Cứu Quốc Tế

Trên thế giới, nhiều người đã nghiên cứu việc dùng học sâu để phát hiện bạo lực từ camera giám sát. Các mô hình như YOLO, SSD hay Faster R-CNN được dùng nhiều trong việc nhận diện đối tượng và hành vi trong video. Đặc biệt, YOLOv11 nhanh và chính xác nên rất hợp cho mấy ứng dụng thời gian thực.

Nhưng mà, các nghiên cứu hiện tại vẫn còn khó khăn như:

* Cảnh quay ở nơi công cộng thường phức tạp.
* Xử lý video thời gian thực đòi hỏi tốc độ cao.
* Phân biệt hành vi bạo lực với hành vi bình thường không dễ.

Những vấn đề này là cơ hội để tối ưu hóa mô hình, cải thiện độ chính xác và tốc độ xử lý.

1.5.2 Hiện Trạng Tại Việt Nam

Ở Việt Nam, học sâu và thị giác máy tính đang phát triển nhanh, nhưng nghiên cứu về phát hiện bạo lực từ camera thì còn ít. Phần lớn mọi người tập trung vào nhận diện đối tượng hay giám sát giao thông, chưa có nhiều ứng dụng cụ thể cho bài toán bạo lực nơi công cộng.

Vậy nên, đề tài này vừa mới mẻ, vừa có tiềm năng áp dụng cao ở Việt Nam, nhất là để tăng cường an ninh ở những chỗ đông người như trường học, bệnh viện hay công viên.

## 1.6 Chiến Lược Tiếp Cận Và Thực Hiện

1.6.1 Hướng Tiếp Cận Tổng Thể

Trước tiên sẽ dùng học sâu, cụ thể là YOLOv11, để phát hiện đối tượng và hành vi bạo lực trong video từ camera. Ngoài ra, Em kết hợp kỹ thuật Pose Estimation để phân tích tư thế của đối tượng, giúp xác định rõ mức độ bạo lực.

1.6.2 Các Bước Thực Hiện Cụ Thể

* Bước 1: Tìm hiểu sâu về YOLOv11 và Pose Estimation.
* Bước 2: Thu thập và xử lý dữ liệu video từ nguồn công cộng và camera thực tế.
* Bước 3: Xây dựng và huấn luyện mô hình phát hiện bạo lực.
* Bước 4: Thử nghiệm hệ thống với video thật và đánh giá kết quả.
* Bước 5: Tối ưu hóa mô hình, đề xuất cải tiến nếu cần.

## 1.7 Kế Hoạch Và Dự Kiến Kết Quả

Cuối cùng sẽ làm theo quy trình từ nghiên cứu lý thuyết, lấy dữ liệu, xây mô hình, đến thử nghiệm và đánh giá. Dự kiến hệ thống sẽ đạt được:

* Một mô hình phát hiện bạo lực chính xác cao, xử lý được video thời gian thực.
* Báo cáo chi tiết về hiệu suất, với mấy chỉ số như accuracy, precision, recall.
* Đề xuất cách áp dụng hệ thống vào các hệ thống an ninh công cộng.

Chương 2. CƠ SỞ LÝ THUYẾT

2.1 TỔNG QUAN VỀ HỌC SÂU

2.1.1 Học sâu (Deep Learning) là gì

Trong những năm gần đây, Trí Tuệ Nhân Tạo (Artificial Intelligence - AI), đặc biệt là Học Máy (Machine Learning), đã trở thành một minh chứng rõ ràng cho cuộc cách mạng công nghiệp lần thứ tư. AI hiện diện trong hầu hết mọi khía cạnh của cuộc sống, từ kinh tế, giáo dục, y tế đến các công việc gia đình, giải trí và thậm chí là lĩnh vực quân sự. Các ứng dụng đáng chú ý của AI xuất hiện từ nhiều lĩnh vực khác nhau để giải quyết các vấn đề đa dạng. Tuy nhiên, phần lớn những đột phá quan trọng lại đến từ Học Sâu (Deep Learning) - một nhánh nhỏ nhưng ngày càng mở rộng và ảnh hưởng đến nhiều loại công việc, từ đơn giản đến phức tạp. Học sâu đã cho phép máy tính thực hiện những nhiệm vụ tưởng chừng như không thể cách đây 15 năm, như phân loại hàng ngàn vật thể trong ảnh, tạo chú thích cho ảnh, bắt chước giọng nói và chữ viết của con người, giao tiếp với con người, và thậm chí là sáng tác văn, phim, ảnh, âm nhạc.

A diagram of machine learning

AI-generated content may be incorrect.

Hình 1

Học sâu là một nhánh của học máy, nhưng trong vòng 5 năm trở lại đây, nó đã trở thành một xu hướng nổi bật trong cuộc cách mạng AI. Có một số lý do chính cho sự phát triển mạnh mẽ này:

* Sự bùng nổ của dữ liệu: Học sâu khai thác hiệu quả dữ liệu lớn (Big Data) với độ chính xác cao hơn so với các phương pháp học máy khác, đặc biệt là trong lĩnh vực xử lý ảnh. Năm 2012, Alex Krizhevsky, Ilya Sutskever và giáo sư hướng dẫn Geoffrey Hinton đã giới thiệu một mô hình gây kinh ngạc cho giới AI, và sau đó là toàn thế giới, khi đạt tỉ lệ lỗi top-5 chỉ 16% trong cuộc thi ILSVRC2012. Đây là lần đầu tiên một mô hình Mạng Nơ-ron Nhân tạo (Artificial Neural Network - ANN) đạt được kết quả tốt nhất trong ngành.
* Phát triển của phần cứng: Sự ra đời của dòng GPU GTX 10 series của NVIDIA vào năm 2014 với hiệu năng tính toán cao và giá thành hợp lý đã làm cho nghiên cứu về Học sâu trở nên dễ tiếp cận hơn, không còn bị giới hạn trong các phòng thí nghiệm đắt đỏ của các trường đại học danh tiếng và các công ty lớn.

Theo thống kê trên trang Papers with Code, hiện có 16 tác vụ lớn mà học máy có thể thực hiện, trong đó hơn một nửa là các tác vụ mà Học sâu đã đạt kết quả tiên tiến nhất (State-of-the-Art - SOTA). Những tác vụ này bao gồm:

* Thị giác máy tính (Computer Vision)
* Xử lý ngôn ngữ tự nhiên (Natural Language Processing)
* Y học (Medical)
* Phương pháp học (Methodology)
* Giọng nói (Speech)
* Chuỗi thời gian (Time Series)
* Âm thanh (Audio)
* Âm nhạc (Music)

2.1.2 Cách hoạt động của DeepLearning:

Học sâu là một phương pháp trong học máy, nơi mạng nơ-ron nhân tạo được thiết kế để mô phỏng khả năng tư duy của bộ não con người. Một mạng nơ-ron gồm nhiều lớp (layer) khác nhau, và số lượng lớp càng nhiều thì mạng càng “sâu”. Mỗi lớp chứa các nút mạng (node) và các nút này được liên kết với các lớp liền kề. Mỗi liên kết giữa các node có một trọng số, và trọng số càng cao thì ảnh hưởng của liên kết đó đến mạng nơ-ron càng lớn.

Mỗi nơ-ron có một hàm kích hoạt, có nhiệm vụ chuẩn hoá đầu ra từ nơ-ron đó. Khi dữ liệu được đưa vào mạng nơ-ron, nó sẽ đi qua tất cả các lớp và trả về kết quả tại lớp cuối cùng, gọi là lớp đầu ra (output layer).

A diagram of machine learning

AI-generated content may be incorrect.

Hình 2

Trong quá trình huấn luyện mô hình mạng nơ-ron, các trọng số này sẽ được điều chỉnh. Mục tiêu của mô hình là tìm ra bộ giá trị trọng số tối ưu để đưa ra các dự đoán chính xác nhất.

Các hệ thống học sâu yêu cầu phần cứng mạnh mẽ để xử lý lượng dữ liệu lớn và thực hiện các phép tính phức tạp. Nhiều mô hình học sâu có thể mất hàng tuần, thậm chí hàng tháng để huấn luyện trên các phần cứng tiên tiến nhất hiện nay..

2.1.3 Ưu nhược điểm của Học sâu:

2.1.3.1 Ưu điểm:

Học sâu đánh dấu một bước ngoặt lớn trong lĩnh vực trí tuệ nhân tạo, cho phép các nhà khoa học dữ liệu xây dựng các mô hình với độ chính xác rất cao trong nhận dạng ảnh, xử lý ngôn ngữ tự nhiên, xử lý giọng nói và nhiều lĩnh vực khác. Một số ưu điểm nổi bật của Học sâu bao gồm:

* Kiến trúc mạng nơ-ron linh hoạt: Có thể dễ dàng điều chỉnh để phù hợp với nhiều vấn đề khác nhau.
* Khả năng giải quyết bài toán phức tạp: Đạt độ chính xác rất cao.
* Tính tự động hóa cao: Có khả năng tự điều chỉnh và tự tối ưu hóa.
* Hiệu năng tốt: Có khả năng thực hiện tính toán song song và xử lý lượng dữ liệu lớn.

2.1.3.2 Nhược điểm:

Mặc dù có nhiều ưu điểm, Học sâu cũng gặp phải một số khó khăn và hạn chế như:

* Yêu cầu dữ liệu lớn: Cần khối lượng dữ liệu rất lớn để phát huy tối đa hiệu quả.
* Chi phí tính toán cao: Do phải xử lý nhiều mô hình phức tạp.
* Thiếu nền tảng lý thuyết mạnh mẽ: Chưa có cơ sở lý thuyết vững chắc để lựa chọn các công cụ tối ưu cho Học sâu.

2.1.4 Một số loại hệ thống học sâu:

Hệ thống học sâu được phân loại theo nhiều cách khác nhau dựa trên cách tổ chức, cách huấn luyện và mục tiêu của chúng. Dưới đây là một số loại hệ thống học sâu phổ biến:

1. Mạng nơ-ron tích chập (Convolutional Neural Networks - CNNs): CNNs chủ yếu được sử dụng trong xử lý hình ảnh và video. Đây là cấu trúc phổ biến nhất cho các bài toán nhận dạng hình ảnh. CNNs sử dụng các lớp tích chập để trích xuất các đặc trưng cục bộ từ ảnh và các lớp gộp để giảm kích thước dữ liệu.
2. Mạng nơ-ron hồi quy (Recurrent Neural Networks - RNNs): RNNs được thiết kế để xử lý dữ liệu tuần tự như chuỗi thời gian hoặc văn bản. Chúng có khả năng ghi nhớ thông tin từ quá khứ thông qua các trạng thái ẩn và được sử dụng rộng rãi trong các bài toán dự đoán chuỗi, dịch máy và sinh văn bản.
3. Mạng nơ-ron hồi quy dài hạn (Long Short-Term Memory - LSTM): LSTM là một dạng đặc biệt của RNNs, được thiết kế để giải quyết vấn đề biến mất gradient trong việc huấn luyện RNNs thông thường. LSTM có khả năng ghi nhớ thông tin trong thời gian dài, làm cho chúng rất hiệu quả trong việc xử lý dữ liệu tuần tự dài.
4. Mạng nơ-ron Gated Recurrent Unit (GRU): Tương tự như LSTM, GRU cũng giải quyết vấn đề biến mất gradient bằng cách sử dụng các cổng. GRU thường được sử dụng khi cần một mạng nơ-ron hồi quy đơn giản hơn so với LSTM.
5. Mạng nơ-ron tự chú ý (Transformer): Transformer ban đầu được giới thiệu cho dịch máy, nhưng nhanh chóng trở thành một trong những mô hình phổ biến nhất cho các ứng dụng xử lý ngôn ngữ tự nhiên (NLP). Transformer sử dụng cơ chế tự chú ý để xử lý các đầu vào không tuần tự một cách hiệu quả.
6. Mạng nơ-ron sinh (Generative Adversarial Networks - GANs): GANs bao gồm hai mạng nơ-ron đối nghịch nhau: một mạng sinh (generator) và một mạng phân biệt (discriminator). Mạng sinh cố gắng tạo ra dữ liệu mới từ dữ liệu ngẫu nhiên, trong khi mạng phân biệt cố gắng phân biệt dữ liệu được tạo ra từ dữ liệu thực tế.
7. Mạng nơ-ron YOLO (You Only Look Once): YOLO là một hệ thống phát hiện vật thể thời gian thực dựa trên kiến trúc mạng nơ-ron tích chập (CNN). Khác với các phương pháp phát hiện vật thể truyền thống (chia thành nhiều giai đoạn), YOLO xử lý toàn bộ hình ảnh chỉ trong một lần duy nhất bằng cách chia ảnh thành các lưới ô vuông và đồng thời dự đoán tọa độ bounding box, độ tin cậy, và lớp đối tượng cho mỗi ô.

2.2 YOLO (You Only Look One)

2.2.1 Giới thiệu về YOLO

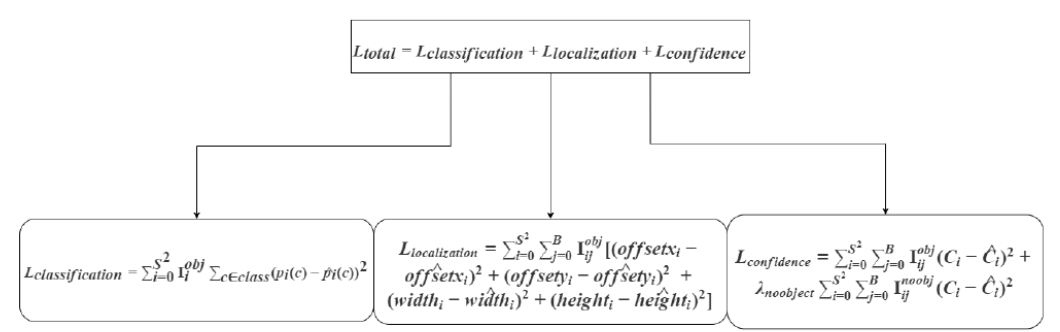
YOLO (You Only Look Once) là một mô hình phát hiện đối tượng theo thời gian thực, được thiết kế dựa trên kiến trúc mạng CNN. Mô hình này có khả năng phát hiện và phân loại đối tượng trong hình ảnh hoặc video. Để đào tạo, cần chuẩn bị tập dữ liệu gồm các hình ảnh biển số xe, trong đó mỗi hình ảnh phải được gắn nhãn với thông tin vị trí và loại đối tượng. Sau đó, lựa chọn phiên bản YOLO phù hợp và xây dựng mạng Nơ-ron CNN để tiến hành quá trình huấn luyện. Quá trình này sử dụng các thuật toán để điều chỉnh trọng số, nhằm giảm sai số dự đoán xuống mức thấp nhất.

2.2.2 Nguyên lý hoạt động trong phát hiện đối tượng

Ở giai đoạn nhận diện, YOLO chia hình ảnh thành mạng lưới các ô SxS, mỗi ô chứa B hộp giới hạn. Mỗi hộp được đặc trưng bởi 5 giá trị:

* (x, y): Tâm hộp so với ô lưới.
* w, h: Chiều rộng và chiều cao hộp (so với toàn bộ hình ảnh).
* Độ tin cậy: Đánh giá khả năng chứa đối tượng thực tế và mức độ chính xác của dự đoán.

YOLO sử dụng các thông số này để xác định vị trí và phân loại đối tượng một cách nhanh chóng và hiệu quả.

****

Hình 2.3

2.2.3 Loss Function của YOLO

* **Classification loss**: tính toán sai số trong việc dự đoán loại nhãn của object. Hàm này chỉ tính trên những ô lưới có xuất hiện object, những ô khác không cần quan tâm
* **Localization loss**: hàm mất mát dùng để tính giá trị lỗi cho bounding box được dự đoán (predicted box) bao gồm x, y, w, h so với vị trí thực tế từ dữ liệu huấn luyện của mô hình (ground truth box). Tại mỗi ô chứa object, ta chọn 1 bounding box có IoU tốt nhất rồi sau đó tính loss theo các bounding box này.
* Confidence loss: hàm mất mát giữa dự đoán bounding box đó có chứa object nào so với label thực tế tại ô vuông đó, được tính trên cả những ô vuông chứa object và không chứa object

2.2.4 Intersection Over Union (IoU)

Intersection Over Union (IoU) là một thông số được sử dụng để đánh giá mức độ chồng lấn giữa hai bounding box trong các bài toán về thị giác máy tính, đặc biệt là trong nhận diện và phát hiện đối tượng. IoU giúp đo lường sự tương đồng giữa bounding box dự đoán và bounding box thực tế (Ground Truth).

Giả sử ta có 2 boxes với thông tin như sau:

* Box 1: có tọa độ top-left là ( x1, y1 ), tọa độ bottom-right là ( x2, y2 ).
* Box 2: có tọa độ top-left là ( x3, y3 ), tọa độ bottom-right là ( x4, y4 ).

Hình 4.2

A diagram of a box

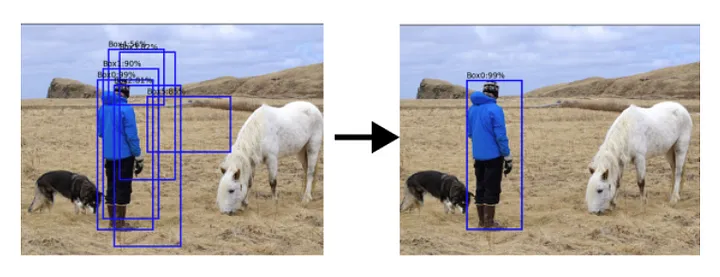
AI-generated content may be incorrect. Khi đó IoU được tính theo công thức:

2.2.5 Thuật toán Non-Maximum Suppression (NMS)

A diagram of a box and a box

AI-generated content may be incorrect.Non-Maximum Suppression (NMS) là một thuật toán quan trọng trong các mô hình phát hiện đối tượng, giúp loại bỏ các bounding box dư thừa và giữ lại các box có độ chính xác cao nhất. Thuật toán này hoạt động dựa trên IoU (Intersection over Union) và confidence score, đảm bảo rằng chỉ những bounding box tốt nhất được giữ lại.

Hình 2.5

****

Hình 2.6

Chi tiết thuật toán NMS

**Input**: Một mảng các bounding box, mỗi box có dạng (x1, y1, x2, y2, c) trong đó:

* (x1, y1) và (x2, y2): Tọa độ góc trên trái và góc dưới phải.
* c: Confidence score do mô hình phát hiện đối tượng cung cấp.
* Ngưỡng IoU: Giá trị dùng để loại bỏ box dư thừa.

**Output**: Một mảng các bounding box sau khi loại bỏ những box trùng lặp.

Các ký hiệu:

* S: bounding box đang xét
* P: Tập các box đầu vào của thuật toán
* thresh\_iou: Ngưỡng IoU để loại bỏ các box thừa
* keep: Tập các box sau khi đã loại bỏ các box thừa

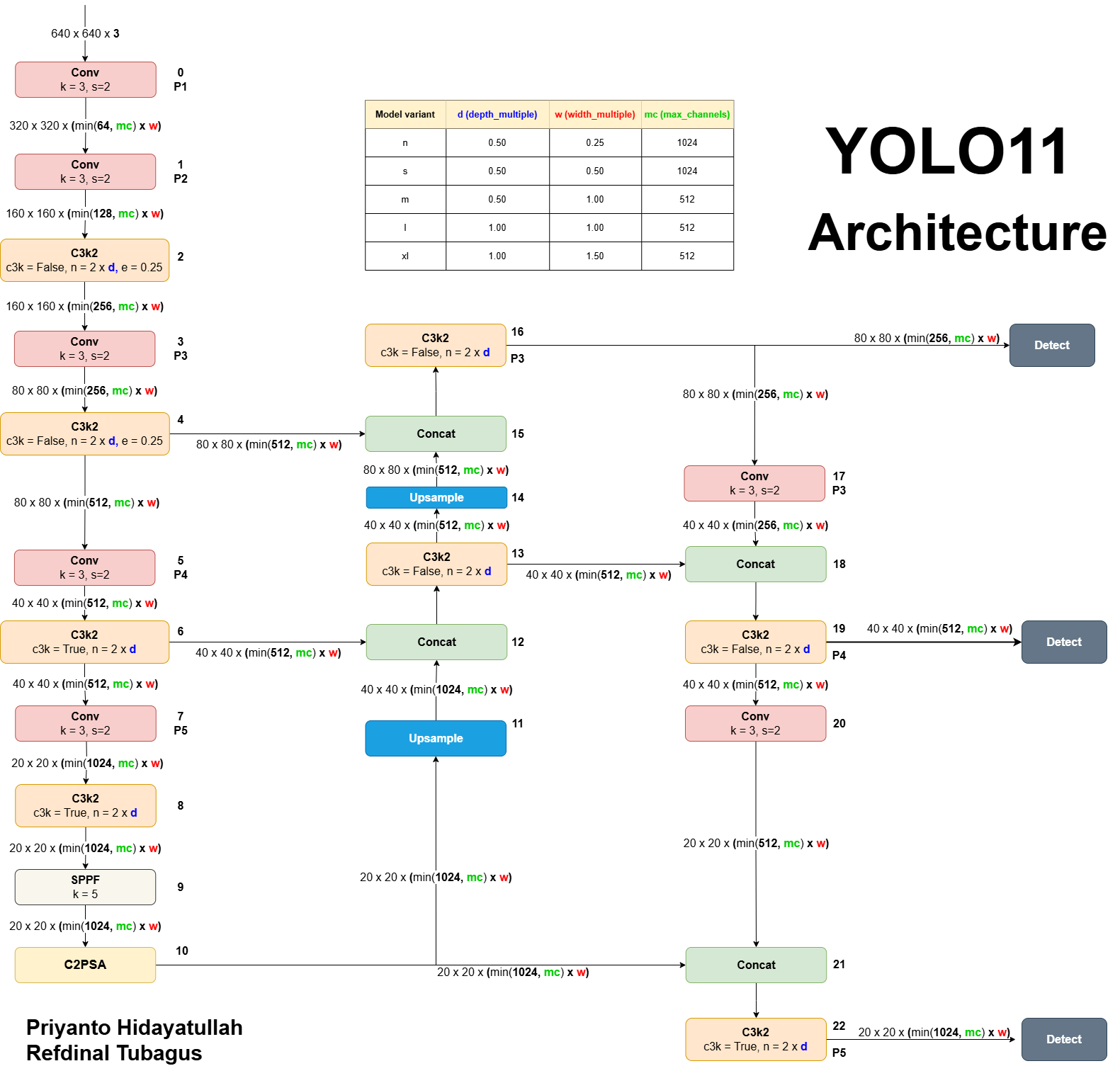
Các bước thực hiện:

**Bước 1**: Chọn box **S** có confidence score cao nhất trong tập **P**, loại bỏ box đó ra khỏi tập **P** và thêm box đó vào tập **keep**.

**Bước 2**: Thực hiện tính toán IOU giữa box **S** vừa lấy ra ở bước 1 với toàn bộ các box còn lại trong tập **P**. Nếu có box nào trong **P** có IOU với box **S** đang xét mà lớn hơn ngưỡng **thresh\_iou** thì loại bỏ box đó ra khỏi **P**

**Bước 3**: Lặp lại bước 1 cho đến khi **P** không còn box nào.

**Kết quả**: Danh sách keep chứa các bounding box sau khi đã loại bỏ các box dư thừa.

2.2.6 Kiến Trúc YOLO11

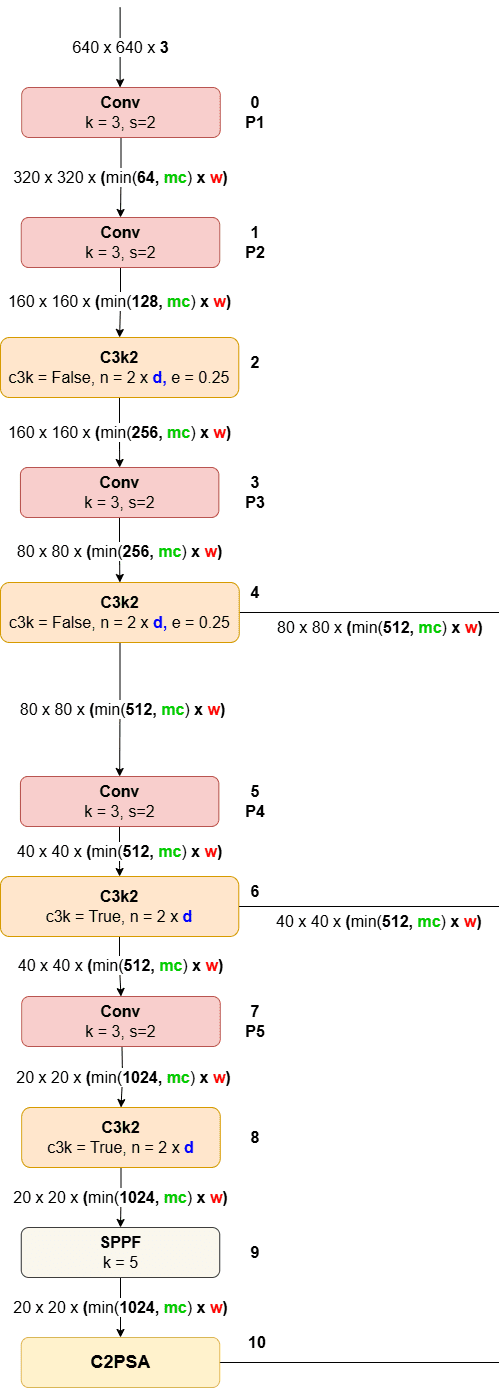
Hình 7: Kiến trúc YOLO11 tổng quát

1. Cách hoạt động của Yolov11

Bao gồm ba phần chính: **Backbone, Neck và Head.**

* **Backbone** là kiến trúc học sâu, đóng vai trò là bộ trích xuất đặc trưng từ ảnh đầu vào.
* **Neck** kết hợp các đặc trưng thu được từ các lớp khác nhau của module Backbone.
* **Head** dự đoán các class và bounding box của các đối tượng là kết quả cuối cùng mà mô hình phát hiện đối tượng tạo ra.

**Backbone:**

Trong **Block 0**, quá trình bắt đầu với kích thước ảnh đầu vào là 640 x 640 x 3, được đưa vào khối tích chập (convolutional block) với kích thước kernel là 3, stride là 2, và padding là 1. Feature map giảm với stride = 2. Khối tích chập tạo ra feature map có kích thước 320 x 320 vì kernel di chuyển theo bước 2 pixel.

Hình 8

Để có được kênh đầu ra của khối tích chập, công thức sau được sử dụng:

min(64, mc) \* w

Trong đó:

* 64 là số kênh đầu ra cơ bản
* mc là max\_channel
* w là width\_multiple

Ví dụ, nếu chúng ta sử dụng biến thể "n" của mô hình YOLOv11, thì số kênh đầu ra cuối cùng sẽ là: min(64, 1024) \* 0.25 = 64 \* 0.25 = 16

Tương tự, phép toán này được tính trong mỗi khối tích chập có mặt trong kiến trúc.

Block 2 là một C3k2 block, chứa hai tham số chính, đó là **c3k** và **n**. Ở đây, **c3k** là tham số boolean cho biết liệu Bottleneck block có sử dụng shortcut hay không. Nếu giá trị của c3k = true, thì Bottleneck block bên trong C3k2 block sẽ sử dụng shortcut, nếu không thì không sử dụng.

Ở đây, **n** xác định số lượng Bottleneck blocks được sử dụng trong C3k2 block. Trong trường hợp của **Block 2**, **n** được tính bằng công thức: n = 2 \* d (depth\_multiple)

Ví dụ, nếu chúng ta sử dụng mô hình YOLOv11 biến thể "n", thì depth\_multiple của mô hình YOLOv11 loại "n" là 0.5, do đó số lượng Bottleneck block sử dụng trong C3k2 block sẽ là:

n = 2 \* 0.5 = 1, tức là 1 Bottleneck block sẽ được sử dụng.

Trong C3k2 block, các Bottleneck blocks này sẽ được xếp chồng lên nhau để tối ưu hóa việc trích xuất đặc trưng từ dữ liệu đầu vào.

Trong **Block 9**, SPPF Block được sử dụng sau lớp convolution cuối cùng của C2f block trong Backbone.

Chức năng chính của SPPF block là tạo ra biểu diễn đặc trưng cố định của đối tượng ở nhiều kích thước khác nhau trong một bức ảnh mà không cần thay đổi kích thước ảnh hoặc làm mất thông tin trong không gian.

**A diagram of a company

AI-generated content may be incorrect.Neck and Head Section**

Hình 9

**Neck Section**: Phần Neck có nhiệm vụ upsampling bản đồ đặc trưng và kết hợp các đặc trưng thu được từ các lớp khác nhau của phần Backbone.

* Lớp upsample trong phần Neck chỉ đơn giản là tăng gấp đôi kích thước của bản đồ đặc trưng mà không thay đổi số lượng kênh đầu ra.
* Concat Block thực hiện phép cộng các kênh đầu ra của các khối được nối lại mà không thay đổi độ phân giải.

**Head Section**: Phần Head chịu trách nhiệm dự đoán các class và bounding box của các đối tượng, đây là đầu ra cuối cùng của mô hình phát hiện đối tượng.

* Detect Block 1 trong phần Head chuyên về việc phát hiện các đối tượng nhỏ, được lấy từ C3k2 block ở Block 16.
* Detect Block 2 trong phần Head chuyên về phát hiện các đối tượng có kích thước trung bình, được lấy từ C3k2 block ở Block 19.
* Detect Block 3 trong phần Head chuyên về phát hiện các đối tượng nhỏ, được lấy từ C3k2 block ở Block 22.

Để có thể đi sâu vào kiến trúc của YOLO11, cần tìm hiểu về các khối (block) cơ bản được sử dụng trong kiến trúc này

1. Convolutional Block (Conv Block)

Bao gồm các lớp Conv2d, BatchNorm2d, và hàm kích hoạt SiLU.

**Lớp Conv2d:**

* Đây là lớp tích chập cơ bản, thực hiện phép toán tích chập giữa một bộ lọc và dữ liệu đầu vào. Quá trình này giúp mô hình trích xuất các đặc trưng từ dữ liệu hình ảnh.
* "Conv2d" nghĩa là phép tích chập được thực hiện trong hai chiều không gian (chiều cao và chiều rộng của ảnh).
* Kích thước đầu ra của lớp Conv2d phụ thuộc vào số lượng bộ lọc (kernels) được sử dụng. Mỗi bộ lọc sẽ tạo ra một feature map riêng biệt, giúp mô hình hiểu rõ hơn các đặc điểm trong ảnh.

**Lớp BatchNorm2d:**

Đây là lớp chuẩn hóa theo batch, giúp ổn định quá trình huấn luyện bằng cách điều chỉnh giá trị của các đặc trưng đầu ra từ lớp Conv2d. Batch normalization giúp làm giảm hiện tượng *vanishing gradients,* giúp mạng học tốt hơn.

**Hàm kích hoạt SILU:**

SiLU (Sigmoid Linear Unit) là hàm kích hoạt được sử dụng trong YOLOv11 thay cho các hàm kích hoạt truyền thống như ReLU. SiLU giúp giới thiệu tính phi tuyến (non-linearity) vào mô hình, cho phép mạng học các đặc trưng phức tạp hơn.

1. Bottleneck Block

**Bottleneck Block** là một khối gồm Conv Block và kết nối shortcut. Khi **shortcut = true**, kết nối shortcut sẽ được áp dụng trong bottleneck block. Ngược lại, nếu không, đầu vào sẽ được truyền qua hai Conv Block liên tiếp.

* **Kết nối Shortcut**: Kết nối này, còn gọi là kết nối bỏ qua hoặc kết nối residual, là một kết nối trực tiếp giúp bỏ qua một hoặc nhiều lớp trong mạng. Mục đích của kết nối shortcut là cho phép gradient dễ dàng truyền qua mạng trong quá trình huấn luyện, giải quyết vấn đề vanishing gradient và giúp mô hình học hiệu quả hơn.
* **Trong bối cảnh Bottleneck Block**: Kết nối shortcut cho phép mô hình bỏ qua các Conv Block nếu cần thiết, thay vào đó sử dụng identity mapping (hoặc bản đồ đồng nhất) mà kết nối shortcut cung cấp. Điều này giúp mô hình dễ dàng học các hàm đồng nhất khi cần thiết, từ đó cải thiện khả năng học các đặc trưng phức tạp và hỗ trợ huấn luyện các mạng nơ-ron sâu, tránh được vấn đề vanishing gradient.

**Vanishing Gradient là gì:**

**Vanishing Gradient** là hiện tượng trong quá trình huấn luyện mạng nơ-ron, khi độ lớn của gradient quá nhỏ, làm cho các trọng số (weights) không thay đổi đáng kể và mô hình không học được hiệu quả, đặc biệt là khi mạng nơ-ron quá sâu.

1. A diagram of a bottle network

   AI-generated content may be incorrect.C2f Block (YOLOv8)

**Hình 2.10 Convolutional Block và Bottle Neck Layer**

**C2f Block** bao gồm một Conv Block, sau đó feature map thu được sẽ được chia làm hai phần. Một phần của feature map đi vào Bottleneck Block, phần còn lại sẽ đi trực tiếp vào Concat Block. Số lượng Bottleneck Block được sử dụng trong C2f Block được xác định bởi tham số depth\_multiple của mô hình. Cuối cùng, feature map từ Bottleneck Block và feature map đã được chia sẻ được nối lại và đưa vào một Conv Block cuối cùng.

1. C3K2 (YOLOv11)

Khác biệt lớn nhất so với YOLOv8 là việc thay thế khối C2f bằng khối C3K2, giúp giảm độ phức tạp tính toán mà vẫn duy trì khả năng nhận diện đối tượng.

C3K2 là khối xử lý đặc trưng chính trong backbone của YOLOv11, kế thừa và cải tiến từ C2F Block của YOLOv8.

Cấu trúc chi tiết:

* Chia feature map đầu vào thành 2 nhánh:
  + Nhánh 1:  
    • Đi qua 2 lớp Conv Block (Conv2d + BatchNorm2d + SiLU) với kernel 3x3.
  + Nhánh 2:  
    • Xử lý qua chuỗi C3K Block (không chia feature map).  
    • C3K Block gồm:
    - 1 Conv Block đầu vào.
    - n lớp Bottleneck (kết hợp Conv Block và shortcut).
    - 1 Conv Block cuối.
* Ghép nối (Concatenate):  
  • Kết hợp đầu ra từ 2 nhánh → Tổng hợp đặc trưng đa chiều.
* A diagram of a diagram

  AI-generated content may be incorrect.Conv Block cuối cùng:  
  • Chuẩn hóa thông tin sau ghép nối → Đầu ra của C3K2 Block.

**Hình 2.11 So sánh giữa C2F và C3K2**

1. Spatial Pyramid Pooling Fast (SPPF) Block

**SPPF Block** gồm một Conv Block và ba lớp MaxPool2d. Các feature map thu được từ mỗi lớp MaxPool2d sẽ được nối lại và đưa vào một Conv Block cuối cùng.

**Spatial Pyramid Pooling (SPP)** là kỹ thuật chia ảnh đầu vào thành nhiều ô và thực hiện pooling trên từng ô. Cách làm này giúp mạng có thể xử lý ảnh với kích thước khác nhau và thu thập thông tin ở nhiều tỷ lệ khác nhau, rất hữu ích cho các tác vụ như nhận dạng đối tượng, nơi các đối tượng có thể xuất hiện ở kích thước khác nhau trong ảnh.

**SPP-Fast** là phiên bản tối ưu của SPP, thay vì sử dụng nhiều mức pooling với các kích thước kernel khác nhau, nó chỉ sử dụng một kernel cố định, giảm bớt tính toán và cải thiện tốc độ, mặc dù có thể ảnh hưởng đôi chút đến độ chính xác.

A diagram of a computer program

AI-generated content may be incorrect.

**Hình 2.12 Spatial Pyramid Pooling Fast**

1. Phần Đặc biệt: C2PSA

YOLOv11 giới thiệu khối C2PSA (Cross Stage Partial with Spatial Attention) – một cải tiến tích hợp cơ chế chú ý không gian, giúp mô hình tập trung vào các vùng quan trọng như vật thể nhỏ hoặc bị che khuất. Khối này tăng cường độ chính xác bằng cách gán trọng số không gian cho các đặc trưng, từ đó làm nổi bật thông tin liên quan trực tiếp đến vật thể cần phát hiện.

Position-Sensitive Attention

Lớp này kết hợp cơ chế chú ý nhạy cảm vị trí và mạng feed-forward để nâng cao khả năng xử lý đặc trưng. Quy trình bao gồm:

* Đầu tiên, lớp Attention phân tích và gán trọng số cho các vùng không gian trong feature map.
* Kết quả từ lớp Attention được ghép nối (concatenate) với feature map gốc, sau đó đưa qua mạng Feed Forward Neural Networks để tối ưu hóa thông tin.
* Dữ liệu tiếp tục đi qua Conv Block có kích hoạt SiLU (gồm Conv2d + BatchNorm2d + SiLU) và Conv Block không kích hoạt (chỉ Conv2d).
* Cuối cùng, đầu ra từ Conv Block được ghép nối với kết quả đã ghép nối ban đầu, tạo thành đầu ra cuối của lớp.

Khối C2PSA

C2PSA kế thừa cấu trúc phân nhánh của C2F Block nhưng thay thế Bottleneck bằng 2 module PSA (Partial Spatial Attention):

* Feature map đầu vào được chia thành 2 nhánh, mỗi nhánh xử lý độc lập bằng một PSA Module. Mỗi PSA Module bao gồm Position-Sensitive Attention và Conv Block.
* Hai nhánh sau đó được ghép nối và đưa qua Conv Block cuối để tổng hợp thông tin.
* Ưu điểm vượt trội:
  + Tập trung vào chi tiết: Cơ chế chú ý không gian giúp mô hình phát hiện vật thể nhỏ hoặc bị che khuất tốt hơn YOLOv8.
  + Hiệu quả tính toán: Dù thêm cơ chế phức tạp, FLOPs chỉ tăng nhờ tối ưu CSP và xử lý song song.
  + A diagram of a system

    AI-generated content may be incorrect.Cân bằng độ chính xác: Duy trì mAP cao ngay cả trong điều kiện ảnh phức tạp (ánh sáng yếu, nhiễu, đông vật thể).

**Hình 2.13 C2-Position Sensitive Attention Block (C2PSA)**

1. Detect Block

**Detect Block** chịu trách nhiệm phát hiện đối tượng trong YOLOv1. Khác với các phiên bản YOLO trước, YOLOv1 là một mô hình anchor-free, có nghĩa là nó dự đoán trực tiếp vị trí trung tâm của đối tượng thay vì dự đoán độ lệch từ một **anchor box** đã biết. Phát hiện không sử dụng anchor giúp giảm số lượng dự đoán hộp, từ đó tăng tốc các bước xử lý hậu kỳ phức tạp để lọc qua các phát hiện ứng viên sau khi suy luận.

**Detect Block** bao gồm hai track. Track đầu tiên dùng để dự đoán bounding box, track thứ hai dùng để dự đoán class. Cả hai track đều có hai **Conv Block**, sau đó là một lớp **Conv2d** duy nhất, đưa ra Bounding Box loss và Class Loss tương ứng.

2.3 RTSP

2.3.1 RTSP là gì?

A close-up of blue cables

AI-generated content may be incorrect.Giao thức Real-Time Streaming Protocol (RTSP) là một giao thức mạng được thiết kế để điều khiển việc phân phối nội dung đa phương tiện qua internet. RTSP hoạt động như một "điều khiển từ xa" cho các máy chủ phát trực tuyến, cho phép người dùng gửi các lệnh như phát, tạm dừng, hoặc dừng để quản lý luồng truyền thông. Được sử dụng phổ biến trong các hệ thống camera IP, hội nghị video và phát sóng trực tuyến, RTSP đóng vai trò quan trọng trong việc cung cấp khả năng điều khiển thời gian thực cho các ứng dụng phát trực tuyến đa phương tiện.

**Hình 14**

2.3.2 Cách RTSP Hoạt động

RTSP thiết lập và duy trì một phiên giữa máy khách và máy chủ. Máy khách gửi các yêu cầu RTSP (như PLAY, PAUSE, hoặc TEARDOWN) để điều khiển luồng truyền thông, trong khi máy chủ phản hồi với các mã trạng thái và thông tin liên quan. Tuy nhiên, RTSP không trực tiếp truyền dữ liệu phương tiện. Thay vào đó, nó phối hợp với:

* **RTP (Real-time Transport Protocol)**: Truyền dữ liệu âm thanh và video.
* **RTCP (Real-time Transport Control Protocol)**: Cung cấp phản hồi về chất lượng truyền dữ liệu.

Quá trình này cho phép RTSP điều khiển phát trực tuyến một cách linh hoạt và hiệu quả, phù hợp với các ứng dụng yêu cầu độ trễ thấp.

2.3.3 Cấu trúc của RTSP

RTSP là một giao thức ứng dụng hoạt động ở tầng ứng dụng của mô hình OSI, được thiết kế để ghép kênh và điều khiển các luồng truyền thông đa phương tiện. Dưới đây là các thành phần chính trong cấu trúc của RTSP:

**Phiên RTSP**

* Một phiên RTSP đại diện cho một kết nối giữa máy khách và máy chủ, bao gồm các luồng truyền thông và thông số điều khiển.
* Mỗi phiên được xác định bởi một **định danh phiên (session ID)** duy nhất.

**Yêu cầu RTSP**

* Máy khách gửi các yêu cầu đến máy chủ để quản lý phiên phát trực tuyến. Các lệnh chính bao gồm:
  + **OPTIONS**: Liệt kê các phương thức mà máy chủ hỗ trợ.
  + **DESCRIBE**: Yêu cầu mô tả phiên (thường sử dụng SDP).
  + **SETUP**: Thiết lập phương thức vận chuyển cho luồng truyền thông (ví dụ: UDP hoặc TCP).
  + **PLAY**: Bắt đầu phát luồng truyền thông.
  + **PAUSE**: Tạm dừng luồng truyền thông.
  + **RECORD**: Ghi lại luồng truyền thông (nếu được hỗ trợ).
  + **TEARDOWN**: Kết thúc phiên và giải phóng tài nguyên.

**Phản hồi RTSP**

* Máy chủ trả lời các yêu cầu bằng mã trạng thái (ví dụ: 200 OK) và thông tin bổ sung, chẳng hạn như định danh phiên hoặc thông số luồng.

**Mô tả phiên (SDP - Session Description Protocol)**

* SDP được sử dụng để trao đổi thông tin về phiên phát trực tuyến giữa máy khách và máy chủ, bao gồm:
  + Loại phương tiện (audio, video).
  + Định dạng mã hóa (codec).
  + Thông số vận chuyển (cổng, giao thức).

**Giao thức vận chuyển**

* RTSP thường kết hợp với:
  + **RTP**: Truyền dữ liệu thời gian thực qua UDP hoặc TCP.
  + **RTCP**: Theo dõi và báo cáo chất lượng dịch vụ (QoS), như độ trễ hoặc mất gói tin.

2.3.4 Ứng dụng của RTSP

Bây giờ, hãy khám phá một số ứng dụng thực tế của RTSP. Một trong những ứng dụng chính của RTSP là trong các hệ thống camera IP được sử dụng cho giám sát và an ninh. Nó cho phép người dùng truy cập các luồng video trực tiếp từ camera từ xa, làm cho nó trở thành một giao thức thiết yếu cho việc giám sát thời gian thực. Ngoài ra, RTSP được sử dụng trong nhiều tình huống phát trực tuyến đa phương tiện khác, chẳng hạn như hội nghị video, phát sóng trực tuyến và các trình phát phương tiện hỗ trợ nội dung phát trực tuyến.

2.3.5 Ưu và nhược điểm của RTSP

**Ưu điểm:**

* **Điều khiển thời gian thực**: Hỗ trợ các lệnh như phát, tạm dừng, tua nhanh, tua lại.
* **Độ trễ thấp:** Lý tưởng cho các ứng dụng yêu cầu phát trực tuyến tức thời.
* **Tính linh hoạt**: Hỗ trợ nhiều định dạng phương tiện và giao thức vận chuyển.
* **Khả năng tương thích:** Hoạt động tốt trong nhiều cấu hình mạng.

**Nhược điểm:**

* **Phức tạp trong cấu hình**: Yêu cầu thiết lập chính xác trên máy chủ và máy khách.
* **Hiệu quả băng thông**: Không tối ưu bằng các giao thức mới như HLS hoặc DASH.
* **Khả năng mở rộng hạn chế**: Gặp khó khăn trong các hệ thống phát trực tuyến quy mô lớn.
* **Hỗ trợ thiết bị**: Không phổ biến trên các trình duyệt và thiết bị di động hiện đại.

## 2.4 CÔNG CỤ ĐƯỢC SỬ DỤNG

2.4.1 Python:

**Python** là một ngôn ngữ lập trình được phát triển bởi Guido van Rossum, có ảnh hưởng từ nhiều ngôn ngữ khác nhau như ABC, Modula 3, Smalltalk và Algol-68. Python đã trải qua nhiều phiên bản, với Python 2.0 và Python 3.0 là hai phiên bản quan trọng nhất. Đây là một ngôn ngữ mã nguồn mở, cho phép bất kỳ ai cũng có thể sử dụng và chỉnh sửa mã nguồn theo nhu cầu của mình.

Python có thể tải xuống và sử dụng miễn phí từ trang chủ của nó, đồng nghĩa với việc mã nguồn của Python có thể được truy cập và điều chỉnh theo yêu cầu dự án. Python được ứng dụng rộng rãi trong nhiều lĩnh vực, bao gồm phát triển web, phân tích dữ liệu, trí tuệ nhân tạo và tự động hóa. Cú pháp đơn giản và dễ đọc của Python là một trong những điểm mạnh của nó, cùng với khả năng tích hợp với nhiều ngôn ngữ khác và sự hỗ trợ từ cộng đồng lập trình viên đông đảo.

A grey background with white arrows

AI-generated content may be incorrect.

Hình 15

**Ưu điểm của Python:**

* Dễ học và dễ đọc: Python có cú pháp rõ ràng và ngắn gọn, giúp người mới học dễ dàng tiếp cận và nhanh chóng nắm bắt ngôn ngữ.
* Đa năng và linh hoạt: Python được sử dụng rộng rãi trong nhiều lĩnh vực như phát triển web, xử lý dữ liệu, học máy, trí tuệ nhân tạo và tự động hóa. Nó cũng có khả năng tích hợp với các ngôn ngữ khác như C/C++ và Java.
* Cú pháp đơn giản và dễ hiểu: Python được thiết kế với cú pháp rõ ràng, giúp mã nguồn dễ đọc, dễ hiểu và dễ bảo trì. Ngôn ngữ này khuyến khích sử dụng khoảng trắng (whitespace) để định dạng mã nguồn, tạo ra cấu trúc mã rõ ràng.
* Hỗ trợ đa nền tảng: Python có thể chạy trên nhiều hệ điều hành khác nhau như Linux, macOS và Windows, cho phép lập trình viên phát triển ứng dụng trên nền tảng họ ưa thích.
* Thư viện và Framework phong phú: Python có một hệ sinh thái phong phú với hàng ngàn thư viện và framework, giúp lập trình viên tiết kiệm thời gian và công sức. Các thư viện như Matplotlib, Requests, SQLAlchemy, Django và Flask là những công cụ quan trọng trong nhiều dự án.
* Cộng đồng lập trình viên mạnh mẽ: Python có một cộng đồng đông đảo và nhiệt tình, luôn sẵn sàng chia sẻ kiến thức và hỗ trợ. Các diễn đàn, blog và nhóm người dùng Python là nguồn tài nguyên quý giá để trao đổi thông tin và giải đáp thắc mắc.
* Hỗ trợ hướng đối tượng (OOP): Python hỗ trợ OOP, cho phép lập trình viên tạo ra các đối tượng và xây dựng các lớp, kế thừa và đa hình. Điều này giúp tăng tính cấu trúc và tái sử dụng mã nguồn, thuận lợi cho việc quản lý và phát triển các dự án lớn.
* Sự bền vững và phát triển liên tục: Python là một ngôn ngữ lập trình bền vững và không ngừng phát triển. Các phiên bản mới được phát hành định kỳ, mang đến những cải tiến và tính năng mới. Tổ chức phát triển Python Software Foundation (PSF) đảm bảo Python sẽ tiếp tục được hỗ trợ và phát triển trong tương lai.

**Nhược điểm của Python:**

* Tốc độ thực thi: So với các ngôn ngữ biên dịch như C++ hay Java, Python có thể chậm hơn vì là ngôn ngữ thông dịch. Tuy nhiên, hiệu suất của Python đã được cải thiện đáng kể nhờ sự phát triển của các trình thông dịch và tối ưu hóa mã nguồn.
* Khả năng mở rộng: Trong một số trường hợp, Python không phải là lựa chọn tốt nhất cho các ứng dụng yêu cầu tài nguyên cao hoặc cần xử lý thời gian thực. Tuy nhiên, Python vẫn có thể tương thích và tích hợp với các thư viện và module hiệu suất cao như C/C++ để tăng cường hiệu suất.

Với sự dễ học, cộng đồng lập trình viên đông đảo và tính linh hoạt trong ứng dụng, Python đã trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên toàn cầu. Điều này chứng tỏ sự đánh giá cao và niềm tin của cộng đồng công nghệ vào Python, cũng như khả năng của nó trong việc phát triển ứng dụng và giải quyết các vấn đề phức tạp một cách hiệu quả. Đáng chú ý, Python là một trong những ngôn ngữ chính thức được sử dụng tại Google. Sự phổ biến và các ưu điểm vượt trội của Python đã thu hút nhiều người sử dụng, từ các lập trình viên chuyên nghiệp đến những người mới bắt đầu học lập trình.

2.4.2 HTML/CSS:

HTML (HyperText Markup Language) và CSS (Cascading Style Sheets) là hai công nghệ cơ bản và quan trọng trong phát triển web. Cả hai đều có vai trò riêng biệt nhưng phối hợp chặt chẽ với nhau để tạo nên các trang web hiện đại và hấp dẫn.

Các điểm chính về HTML và CSS:

* Cấu trúc và trình bày: HTML sử dụng các thẻ (tags) để định nghĩa các thành phần của trang web như đoạn văn, tiêu đề, hình ảnh, liên kết, trong khi CSS quản lý cách các phần tử HTML được hiển thị trên trang web, bao gồm màu sắc, phông chữ, bố cục và khoảng cách.
* Đa dạng và linh hoạt: HTML có thể nhúng các đối tượng đa phương tiện như video, âm thanh và hình ảnh vào trang web. CSS cung cấp các công cụ mạnh mẽ để tạo ra các thiết kế phức tạp và linh hoạt, từ các hiệu ứng đơn giản đến các hiệu ứng động phức tạp.
* Tương thích: HTML và CSS tương thích với nhiều trình duyệt web và các thiết bị khác nhau, từ máy tính để bàn đến điện thoại di động.

**Ưu điểm của HTML và CSS:**

* Dễ học và sử dụng: HTML có cú pháp đơn giản và dễ hiểu, làm cho việc học và sử dụng HTML trở nên dễ dàng. CSS cung cấp các công cụ mạnh mẽ để tạo ra các thiết kế phức tạp và linh hoạt.
* Độc lập nền tảng: HTML và CSS có thể chạy trên mọi nền tảng và trình duyệt mà không cần phải thay đổi mã nguồn.
* Tăng tốc độ tải trang: Bằng cách sử dụng CSS để định kiểu, trang web có thể tải nhanh hơn do giảm kích thước mã nguồn HTML.
* Cộng đồng và tài liệu phong phú: Có một cộng đồng lớn và nhiều tài liệu hỗ trợ, giúp người học và nhà phát triển dễ dàng tìm kiếm và giải quyết các vấn đề liên quan đến HTML và CSS.

**Nhược điểm của HTML và CSS:**

* Giới hạn trong thiết kế: HTML chỉ tạo cấu trúc và nội dung, không thể quản lý các yếu tố liên quan đến thiết kế và bố cục trang web mà cần có sự hỗ trợ của CSS.
* Khó kiểm soát trên các trình duyệt cũ: Một số trình duyệt cũ có thể không hỗ trợ đầy đủ các tính năng mới của CSS, gây ra sự không đồng nhất trong hiển thị.
* Cần nhiều thời gian để thành thạo: Mặc dù cú pháp CSS không quá phức tạp, nhưng việc sử dụng hiệu quả các tính năng nâng cao của CSS có thể đòi hỏi nhiều thời gian và kinh nghiệm.

HTML và CSS là hai ngôn ngữ cốt lõi trong phát triển web. Trong khi HTML cung cấp cấu trúc và nội dung cho trang web, CSS mang lại khả năng định dạng và thiết kế, tạo ra các trang web hấp dẫn và thân thiện với người dùng. Sự kết hợp của HTML và CSS là nền tảng cho mọi dự án web, từ những trang web đơn giản đến những ứng dụng web phức tạp.

2.4.3 OpenCV trong Python:

OpenCV (Open Source Computer Vision Library) là một thư viện mã nguồn mở, được khởi xướng bởi nhóm nghiên cứu tại Intel vào năm 1999. Thư viện này cung cấp nhiều công cụ và thuật toán phục vụ cho xử lý ảnh, học máy và thị giác máy tính, đồng thời hỗ trợ tăng tốc GPU để xử lý dữ liệu theo thời gian thực. OpenCV tương thích với nhiều ngôn ngữ lập trình như C++, C, Python và Java, và hoạt động trên các hệ điều hành như Windows, Linux, macOS, iOS và Android.

OpenCV được phát hành theo giấy phép BSD, cho phép sử dụng và phát triển với mục đích thương mại. Nhờ cộng đồng người dùng rộng lớn, OpenCV đã trở thành một công cụ quan trọng trong nhiều lĩnh vực như xử lý hình ảnh đường phố, giám sát tự động, robot, xe tự hành, phân tích hình ảnh y học, và tìm kiếm và phục hồi hình A black and grey logo

AI-generated content may be incorrect.ảnh/video.

Hình 16

**Ưu điểm của OpenCV:**

* Hỗ trợ đa nền tảng và đa ngôn ngữ: OpenCV tương thích với nhiều ngôn ngữ lập trình và hệ điều hành, giúp lập trình viên dễ dàng tích hợp vào các dự án.
* Tính năng phong phú: Thư viện cung cấp một loạt các công cụ và thuật toán cho xử lý ảnh và thị giác máy tính, từ các thao tác cơ bản đến các thuật toán phức tạp.
* Tăng tốc GPU: OpenCV có khả năng sử dụng GPU để tăng tốc độ xử lý, giúp thực hiện các tác vụ thời gian thực hiệu quả hơn.
* Miễn phí và mã nguồn mở: Được phát hành theo giấy phép BSD, OpenCV cho phép sử dụng tự do trong cả các dự án thương mại và phi thương mại.

OpenCV được tổ chức theo cấu trúc module, bao gồm các thư viện liên kết tĩnh hoặc động. Một số module phổ biến bao gồm:

* Core functionality (core): Chứa các chức năng cơ bản và các loại cấu trúc dữ liệu như mảng đa chiều dày đặc.
* Image Processing (imgproc): Cung cấp các công cụ xử lý ảnh như lọc tuyến tính và phi tuyến, biến đổi hình học, thay đổi kích thước và góc, chuyển đổi không gian màu và tạo biểu đồ.
* Video Analysis (video): Dùng cho phân tích video như tách nền, theo dõi vật thể và ước tính chuyển động.
* Object Detection (objdetect): Dùng để nhận diện và mô phỏng các đối tượng như khuôn mặt, con người và xe hơi.
* Camera Calibration and 3D Reconstruction (calib3d): Cung cấp các thuật toán hình học đa chiều cơ bản, ước tính kiểu dáng đối tượng, thuật toán thư tín âm thanh nổi và tái tạo 3D, hiệu chuẩn máy ảnh đơn và stereo.
* High-level GUI (highgui): Cung cấp giao diện đồ họa cấp cao để tương tác với người dùng.
* Video I/O (videoio): Cung cấp giao diện để thu và mã hóa video.

Ngoài ra, OpenCV còn có các module hỗ trợ khác như Python binding, FLANN và Google test wrapper.

Với các tính năng và ưu điểm nổi bật, OpenCV đã trở thành một công cụ không thể thiếu trong lĩnh vực xử lý ảnh và thị giác máy tính, hỗ trợ các nhà phát triển và nghiên cứu giải quyết các bài toán phức tạp một cách hiệu quả.

2.4.3.1 Ứng dụng của OpenCV trong Python:

Trong Python, OpenCV là một thư viện mạnh mẽ và dễ sử dụng cho các nhiệm vụ xử lý ảnh và thị giác máy tính. Dưới đây là một số ứng dụng phổ biến của OpenCV trong Python:

* Xử lý ảnh: Các thao tác cơ bản như cắt, thay đổi kích thước, và lọc ảnh.
* Phát hiện và nhận diện đối tượng: Sử dụng các mô hình học máy và Học sâu để nhận diện khuôn mặt, con người, xe hơi và các đối tượng khác.
* Theo dõi đối tượng: Theo dõi chuyển động của các đối tượng trong video theo thời gian thực.
* Hiệu chỉnh máy ảnh: Hiệu chuẩn máy ảnh và tái tạo hình ảnh 3D từ các cặp ảnh stereo.
* Phân tích video: Tách nền, phát hiện chuyển động, và phân tích hành vi từ video.

2.4.4 Google Colab:

Google Colab, hay Colaboratory, là một môi trường lập trình trực tuyến được phát triển bởi Google, cho phép người dùng viết và thực thi mã Python trực tiếp trên trình duyệt. Được xây dựng trên nền tảng Jupyter Notebook, Google Colab cung cấp các công cụ mạnh mẽ để hỗ trợ học máy, phân tích dữ liệu, và nghiên cứu khoa học.

Các tính năng nổi bật của Google Colab

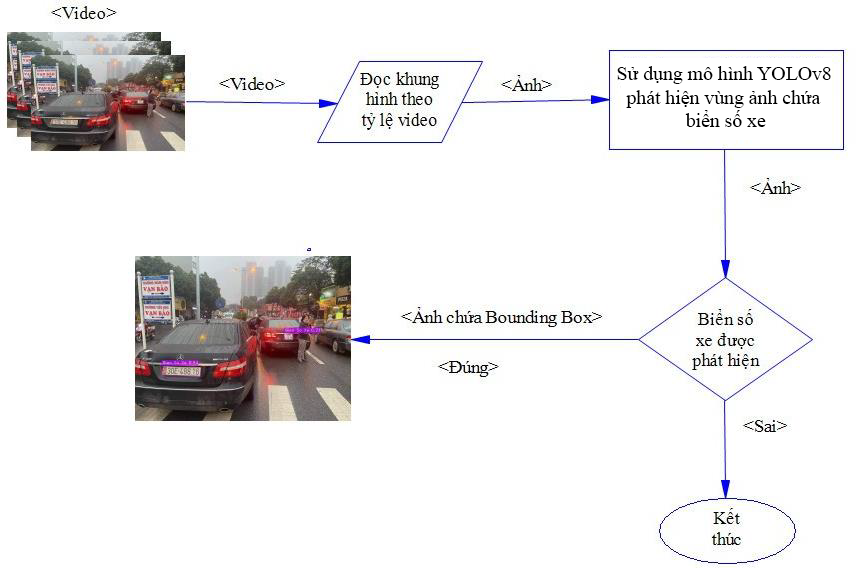
* Miễn phí và dễ sử dụng: Google Colab hoàn toàn miễn phí và không yêu cầu cài đặt phần mềm. Người dùng chỉ cần một tài khoản Google để bắt đầu sử dụng.
* Tài nguyên mạnh mẽ: Colab cung cấp truy cập miễn phí tới các tài nguyên tính toán mạnh mẽ như GPU và TPU, giúp tăng tốc quá trình huấn luyện mô hình học máy.
* Hỗ trợ thư viện phong phú: Colab tích hợp sẵn nhiều thư viện phổ biến trong học máy và phân tích dữ liệu như TensorFlow, PyTorch, Keras, NumPy, Pandas, Matplotlib, và nhiều hơn nữa.
* Chia sẻ và cộng tác: Người dùng có thể dễ dàng chia sẻ notebook của mình với người khác và cùng làm việc trên cùng một tài liệu, tương tự như Google Docs.
* Lưu trữ trên đám mây: Colab lưu trữ notebook trên Google Drive, cho phép truy cập và quản lý dễ dàng từ bất kỳ thiết bị nào có kết nối internet.
* Tích hợp với GitHub: Người dùng có thể mở trực tiếp các notebook từ GitHub, chỉnh sửa và lưu lại các thay đổi.

Google Colab được sử dụng rộng rãi trong nhiều lĩnh vực, bao gồm:

* Học máy và trí tuệ nhân tạo: Tận dụng sức mạnh tính toán của GPU và TPU, Colab là công cụ lý tưởng để huấn luyện các mô hình học máy và AI.
* Phân tích dữ liệu: Colab hỗ trợ các nhà khoa học dữ liệu trong việc xử lý, phân tích và trực quan hóa dữ liệu một cách hiệu quả.
* Nghiên cứu khoa học: Các nhà nghiên cứu có thể sử dụng Colab để viết và chạy các mô phỏng, phân tích số liệu và chia sẻ kết quả nghiên cứu.
* Giáo dục: Colab là công cụ tuyệt vời cho giáo viên và học sinh trong việc giảng dạy và học tập các khái niệm về lập trình, học máy và phân tích dữ liệu.

Google Colab là một công cụ mạnh mẽ và tiện lợi cho lập trình Python trực tuyến, cung cấp tài nguyên tính toán miễn phí và hỗ trợ nhiều thư viện phổ biến. Với tính năng chia sẻ và cộng tác dễ dàng, Colab đã trở thành lựa chọn hàng đầu cho các nhà phát triển, nhà nghiên cứu và học viên trong nhiều lĩnh vực khác nhau.

CHƯƠNG 3. XÂY DỰNG CÔNG CỤ PHÁT HIỆN BẠO LỰC

3.1 Mô hình bài toán

Hình 17: Mô hình bài toán

Hình trên thể hiện mô hình bài toán thực hiện việc nhận diện biển số xe của các mô hình YOLOv8. Dữ liệu cần kiểm tra sẽ được thu thập từ nhiều nguồn khác nhau có thể là ảnh hoặc video. Dữ liệu sẽ được đưa vào mô hình, đối với video sẽ thực hiện việc đọc khung hình để đưa ra ảnh có pixel phù hợp được quy định. Sử dụng mô hình YOLOv8 để nhận diện ảnh có chứa biển số xe, nếu ảnh có chứa biển số xe mô hình sẽ khoanh vùng bằng Bounding Box và đưa ra hình ảnh hoặc video tùy thuộc theo dữ liệu đầu vào; nếu ảnh không chứa biển số xe mô hình sẽ kết thúc.

3.2 Chuẩn bị dữ liệu

**Cách Image Annotation hoạt động**

Để label cho hình ảnh, bạn có thể sử dụng bất kỳ công cụ label dữ liệu nào mã nguồn mở hoặc miễn phí. **Computer Vision Annotation Tool (CVAT)** có lẽ là công cụ label hình ảnh mã nguồn mở phổ biến nhất hiện nay.

Khi xử lý một lượng lớn dữ liệu, bạn sẽ cần một đội ngũ được đào tạo để label cho các hình ảnh. Các công ty thường sử dụng chính các nhà khoa học dữ liệu của mình để label, nhưng đối với các dự án phức tạp và thực tế hơn, họ thường phải thuê các nhà cung cấp dịch vụ label video AI chuyên nghiệp.

Các công cụ label cung cấp nhiều tính năng khác nhau để label hiệu quả cho một hoặc nhiều khung hình. Nhãn (labels) được áp dụng cho các đối tượng trong hình ảnh bằng nhiều kỹ thuật khác nhau, và số lượng nhãn trên mỗi hình ảnh có thể thay đổi tùy thuộc vào yêu cầu của từng ứng dụng cụ thể.

**Cách label hình ảnh**

Quy trình label hình ảnh thường được thực hiện theo các bước sau:

**Bước 1: Chuẩn bị dataset**

Thu thập và tổ chức hình ảnh vào một thư mục hoặc cấu trúc dễ quản lý. Đảm bảo rằng hình ảnh có chất lượng tốt và phù hợp với mục tiêu dự án.

**Bước 2: Xác định các class labels cần phát hiện**

Trước khi bắt đầu label, bạn cần liệt kê các class labels mà mô hình sẽ nhận diện (ví dụ: xe ô tô, người, cây cối, v.v.).

**Bước 3: Vẽ bounding box xung quanh đối tượng bạn muốn phát hiện trong mỗi hình ảnh**

Sử dụng các công cụ như hình chữ nhật (rectangle), đa giác (polygon), hoặc các hình dạng khác để khoanh vùng đối tượng cần label.

**Bước 4: Chọn class label cho mỗi bounding box bạn đã vẽ**

Sau khi vẽ xong, bạn cần chỉ định class label cho từng đối tượng (ví dụ: nếu bạn đã vẽ một hộp quanh một biển số xe, hãy label đó là "license plate").

**Bước 5: Xuất các annotations theo định dạng yêu cầu**

Sau khi hoàn tất label, xuất dữ liệu thành các định dạng phổ biến như COCO JSON, YOLO, Pascal VOC, hoặc các định dạng khác tùy thuộc vào yêu cầu của mô hình học máy bạn sẽ sử dụng.

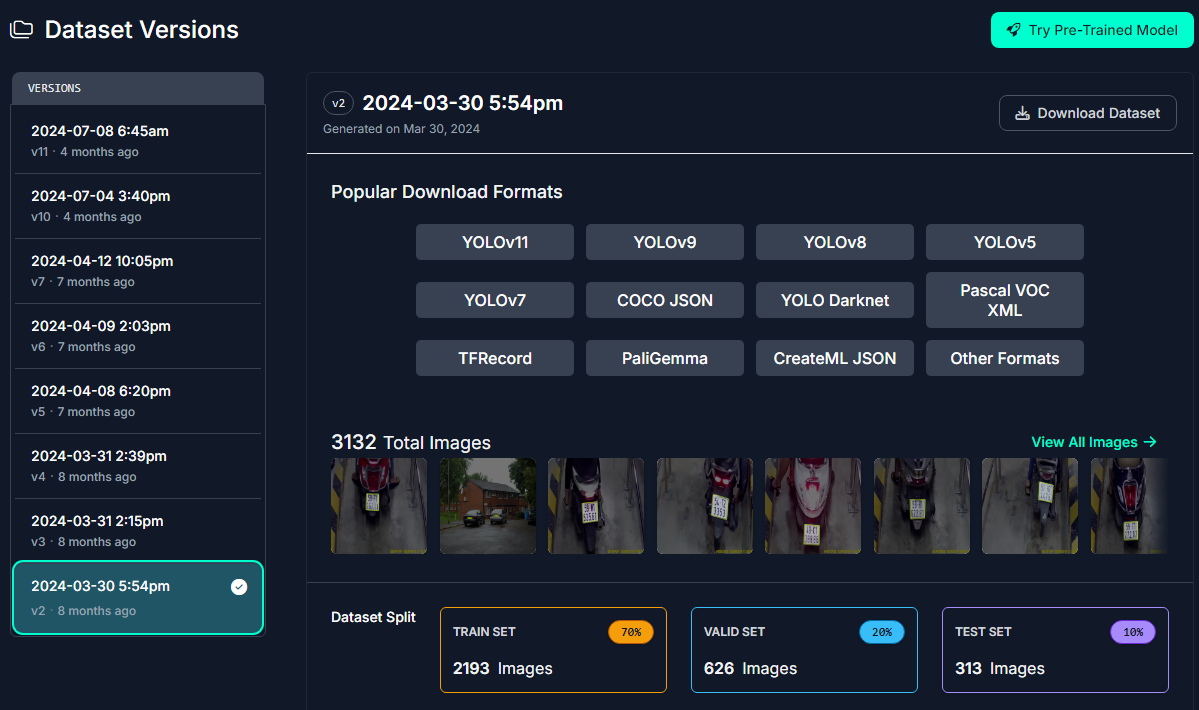


3.3 Dataset

3.3.1 Giới thiệu về Dataset

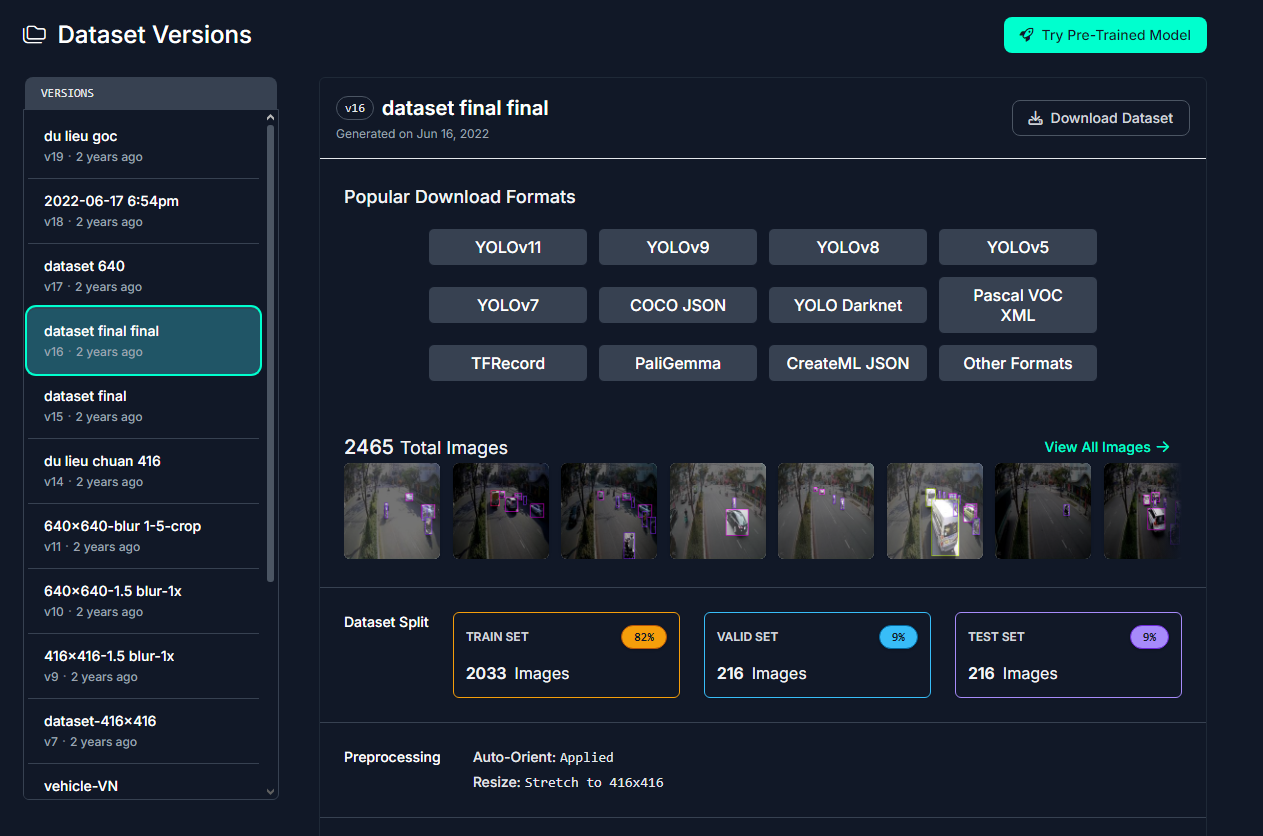
Ứng dụng phát hiện và nhận diện biển số xe kết hợp phát hiện phương tiện giao thông sử dụng hai dataset độc lập. Mục tiêu của việc sử dụng hai loại dataset này là đảm bảo khả năng phát hiện đồng thời phương tiện giao thông và biển số xe trong thực tế. Cả hai tập dữ liệu đều được tải xuống từ nền tảng Roboflow Universe, một nguồn cung cấp tập dữ liệu chất lượng cao dành cho các ứng dụng thị giác máy tính.

3.3.2 Dataset Biển Số Xe

****

* Nguồn: License Plate Recognition Dataset trên Roboflow.
* Mục đích: Được sử dụng để phát hiện và nhận diện biển số xe.
* Số lượng mẫu: 3132 hình ảnh.
* Kích thước hình ảnh: Tất cả hình ảnh đã được chuẩn hóa kích thước về 640x640.
* Phân chia dữ liệu:
* Tập huấn luyện: 2193 ảnh (70%).
* Tập xác thực: 626 ảnh (20%).
* Tập kiểm tra: 313 ảnh (10%).

3.3.3 Dataset Phương Tiện Giao Thông

****

* Nguồn: Vehicle Vietnam Dataset trên Roboflow.
* Mục đích: Được sử dụng để phát hiện các loại phương tiện giao thông như xe máy, ô tô, xe tải, v.v.
* Số lượng mẫu: 2465 hình ảnh.
* Kích thước hình ảnh: Tất cả hình ảnh đã được chuẩn hóa kích thước về 320x320.
* Phân chia dữ liệu:
* Tập huấn luyện: 2033 ảnh (80%).
* Tập xác thực: 216 ảnh (10%).
* Tập kiểm tra: 216 ảnh (10%).

3.4 Huấn luyện và đánh giá mô hình

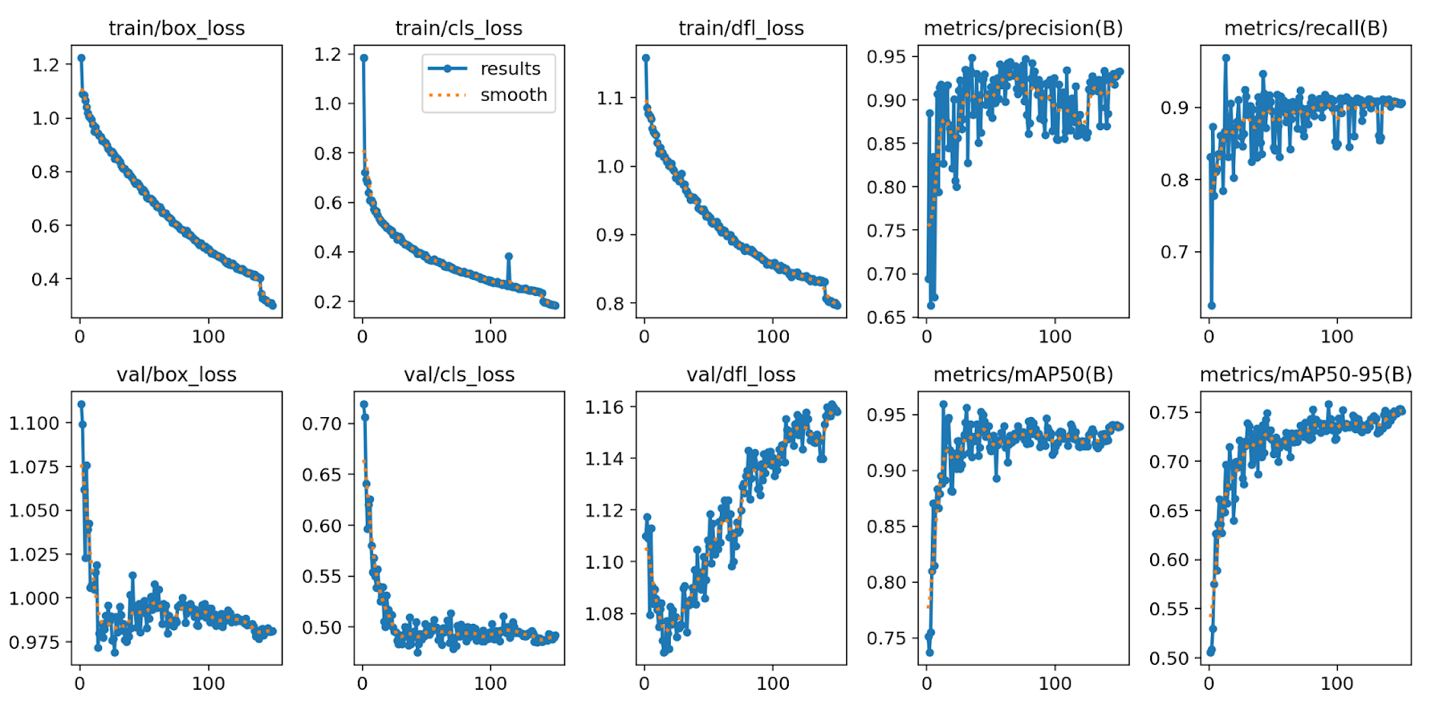
3.4.1 Huấn luyện YOLOv8 trong việc phát hiện xe

Mô hình YOLOv8 được huấn luyện trong 150 epochs với cấu hình ảnh đầu vào (image size - imgsz) cố định là 640. Dữ liệu được chia theo tỷ lệ: 80% (2033 imgs) dành cho tập huấn luyện (training), 10% (216 imgs) dành cho tập xác thực (validation), và 10% (216 imgs) còn lại dành cho tập kiểm tra (test).

Quá trình huấn luyện tập trung vào việc tối ưu các thành phần mất mát (loss) bao gồm:

* **Box Loss:** đánh giá độ chính xác của khung giới hạn dự đoán so với thực tế.
* **Classification Loss:** đánh giá độ chính xác của phân loại đối tượng.
* **DFL Loss:** tối ưu hóa dự đoán phân phối định lượng (Distribution Focal Loss).

Các chỉ số hiệu năng chính, bao gồm Precision (P), Recall (R), mAP@50 và mAP@50-95, được tính toán trên tập kiểm tra sau mỗi epoch nhằm đánh giá độ chính xác và tính ổn định của mô hình.

****

Hình 18: Đồ thị mất mát và các chỉ số hiệu năng trong quá trình huấn luyện

**Đánh giá mô hình**

Dựa trên đồ thị trong Hình 1, quá trình huấn luyện cho thấy xu hướng giảm dần đều của các giá trị mất mát (loss) theo số lượng epoch, cụ thể:

* Box Loss (train/val): Giảm ổn định từ khoảng 1.2 xuống dưới 0.4, cho thấy khả năng cải thiện liên tục của mô hình trong việc dự đoán chính xác vị trí khung giới hạn.
* Classification Loss (train/val): Giảm đáng kể từ giá trị ban đầu khoảng 1.2 xuống dưới 0.2, phản ánh sự cải thiện trong khả năng phân loại đối tượng biển số xe.
* DFL Loss (train/val): Giảm nhẹ và ổn định, đạt giá trị dưới 0.8 sau khi hoàn thành huấn luyện. Validation có xu hướng tăng dần ở giai đoạn cuối, cho thấy khả năng overfitting nhẹ.

Ngoài ra, các chỉ số đánh giá hiệu năng như Precision, Recall, mAP@50, và mAP@50-95 cho thấy sự cải thiện rõ rệt qua từng epoch, phản ánh khả năng học sâu và ổn định của mô hình.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Class** | **Images** | **Instance** | **Precision(P)** | **Recall (R)** | **mAP@50** | **mAP@50-95** |
| all | 216 | 1265 | 0.919 | 0.904 | 0.947 | 0.758 |
| bus | 4 | 6 | 1 | 0.792 | 0.904 | 0.788 |
| car | 192 | 614 | 0.959 | 0.971 | 0.983 | 0.824 |
| motorbike | 203 | 607 | 0.893 | 0.906 | 0.953 | 0.641 |
| truck | 34 | 38 | 0.825 | 0.947 | 0.948 | 0.779 |

Các chỉ số cho thấy mô hình đạt độ chính xác cao trên tập test:

* **Precision (P)**: 91.9%, đảm bảo mô hình nhận diện đúng các biển số xe mà không dự đoán dư thừa.
* **Recall (R)**: 90.4%, thể hiện khả năng tìm kiếm hầu hết các biển số xe thực tế trong tập test.
* **mAP@50**: 94.7%, chỉ số đánh giá chính xác của mô hình ở mức IoU=0.5.
* **mAP@50-95**: 75.8%, đo độ chính xác trung bình trên nhiều ngưỡng IoU (từ 0.5 đến 0.95), phản ánh tính ổn định và hiệu quả cao của mô hình trên nhiều ngưỡng khác nhau.

**Phân Tích Confusion Matrix**

Ma trận nhầm lẫn được minh họa trong Hình 2 cung cấp cái nhìn sâu hơn về khả năng phân loại của mô hình. Các kết quả chính bao gồm:

**Car:**

* Dự đoán chính xác: 596 trường hợp.
* Sai sót: 31 trường hợp bị nhầm là "Background" và một số nhỏ nhầm lẫn với các lớp khác.

**Motorbike:**

* Dự đoán chính xác: 561 trường hợp.
* Sai sót: 75 trường hợp bị nhầm là "Background".

**Truck (Xe tải):**

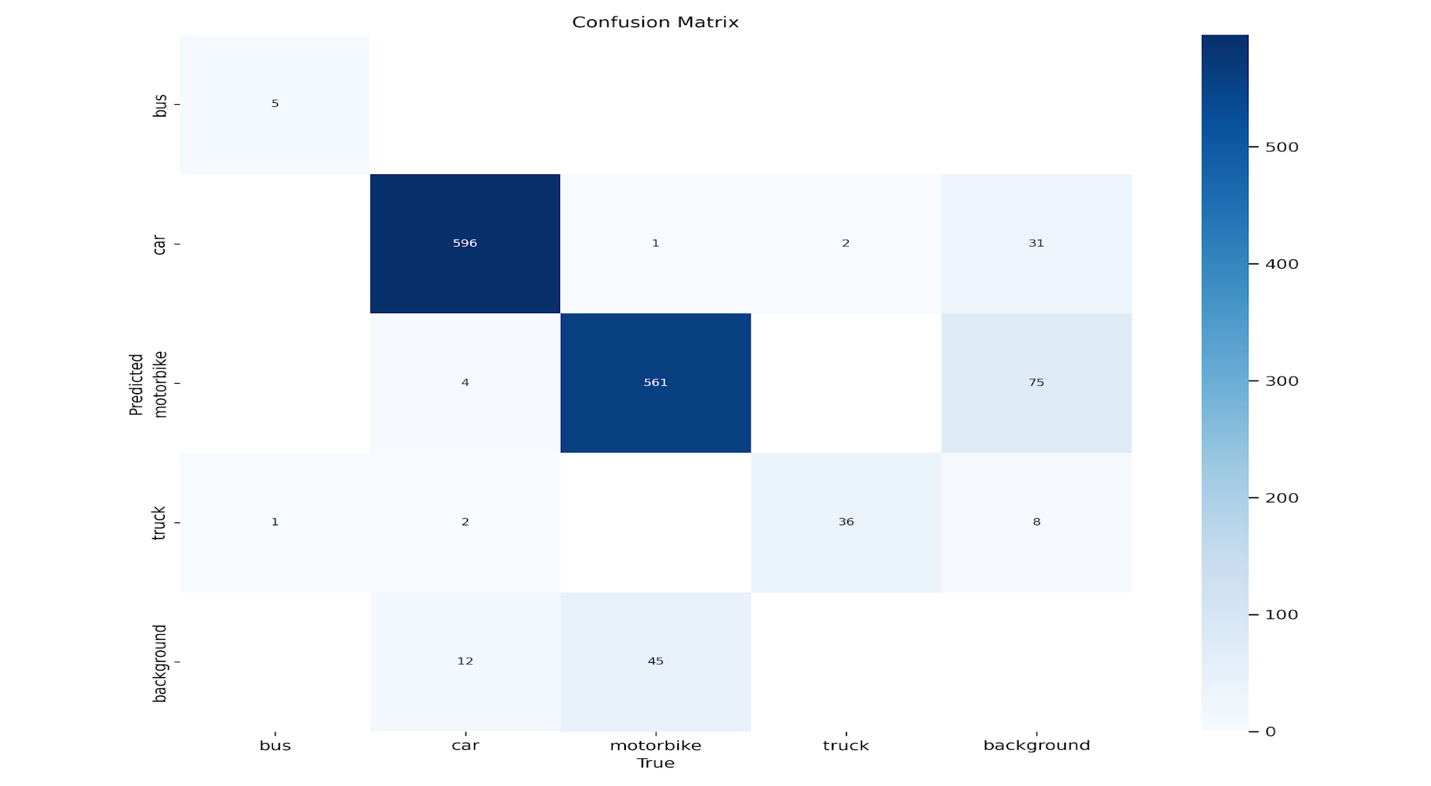
* Dự đoán chính xác: 36 trường hợp.
* Sai sót: 8 nhầm là "Background".

**Bus:**

* Dự đoán chính xác: 5 trường hợp, ít dữ liệu trong lớp này dẫn đến khó phân loại chính xác.

**Background (Nền):**

* Dự đoán chính xác: Hầu hết các nhầm lẫn đến từ các lớp khác, ví dụ 45 nhầm từ "Car".

****

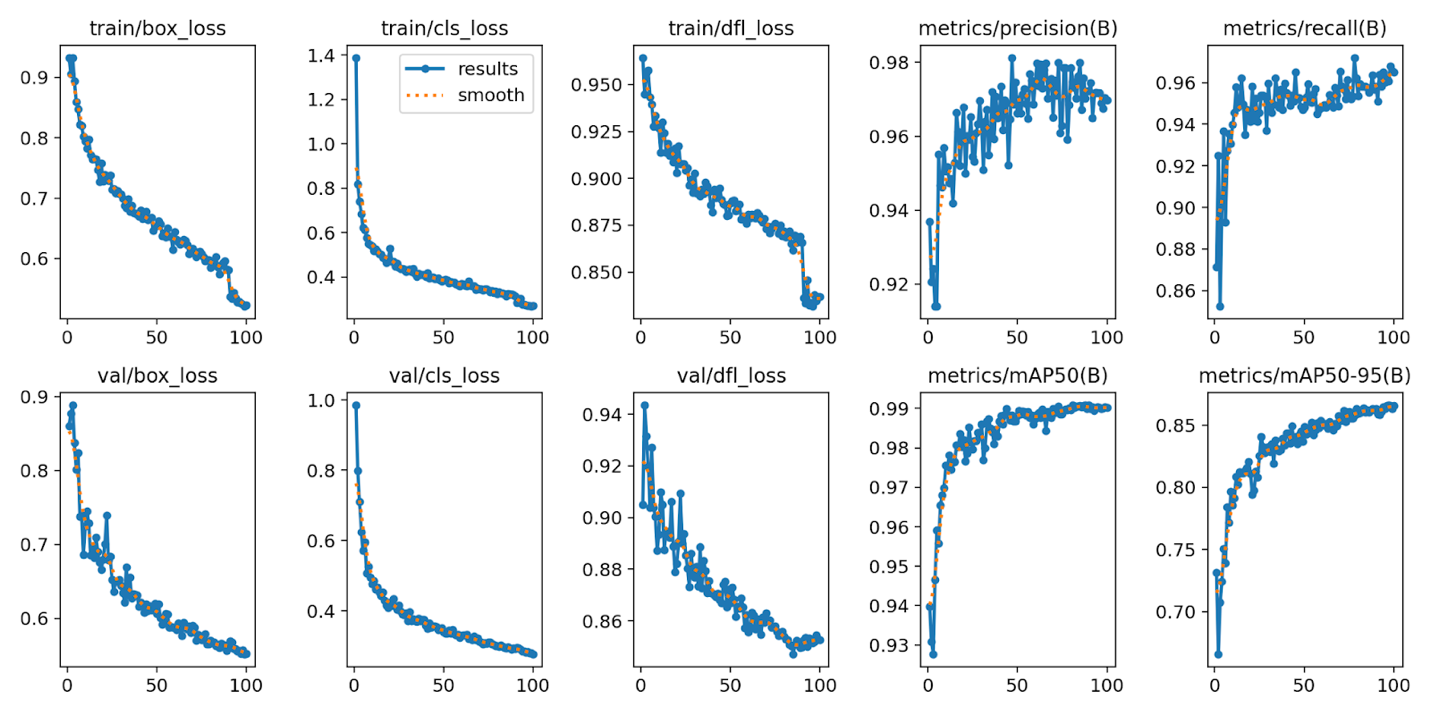
Hình 19: Confusion Matrix trên tập kiểm tra

3.4.2 Huấn luyện YOLOv8 trong việc phát hiện biển số xe

Mô hình YOLOv8 được huấn luyện trong 100 epochs với cấu hình ảnh đầu vào (image size - imgsz) cố định là 320. Dữ liệu được chia theo tỷ lệ: 70% (2832 imgs) dành cho tập huấn luyện (training), 20% (813 imgs) dành cho tập xác thực (validation), và 10% (398 imgs) còn lại dành cho tập kiểm tra (test).

Quá trình huấn luyện tập trung vào việc tối ưu các thành phần mất mát (loss) bao gồm:

* Box Loss: đánh giá độ chính xác của khung giới hạn dự đoán so với thực tế.
* Classification Loss: đánh giá độ chính xác của phân loại đối tượng.
* DFL Loss: tối ưu hóa dự đoán phân phối định lượng (Distribution Focal Loss).

Các chỉ số hiệu năng chính, bao gồm Precision (P), Recall (R), mAP@50 và mAP@50-95, được tính toán trên tập kiểm tra sau mỗi epoch nhằm đánh giá độ chính xác và tính ổn định của mô hình**Đánh giá mô hình**

Hình 20: Đồ thị mất mát và các chỉ số hiệu năng trong quá trình huấn luyện

Dựa trên đồ thị trong Hình 1, quá trình huấn luyện cho thấy xu hướng giảm dần đều của các giá trị mất mát (loss) theo số lượng epoch, cụ thể:

* Box Loss (train/val): Giảm ổn định từ khoảng 0.9 xuống dưới 0.6, cho thấy khả năng cải thiện liên tục của mô hình trong việc dự đoán chính xác vị trí khung giới hạn.
* Classification Loss (train/val): Giảm đáng kể từ giá trị ban đầu khoảng 1.4 xuống dưới 0.4, phản ánh sự cải thiện trong khả năng phân loại đối tượng biển số xe.
* DFL Loss (train/val): Giảm nhẹ và ổn định, đạt giá trị dưới 0.85 sau khi hoàn thành huấn luyện.

Ngoài ra, các chỉ số đánh giá hiệu năng như Precision, Recall, mAP@50, và mAP@50-95 cho thấy sự cải thiện rõ rệt qua từng epoch, phản ánh khả năng học sâu và ổn định của mô hình.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Class** | **Images** | **Instance** | **Precision(P)** | **Recall (R)** | **mAP@50** | **mAP@50-95** |
| LicensePlate | 626 | 715 | 0.969 | 0.961 | 0.99 | 0.866 |

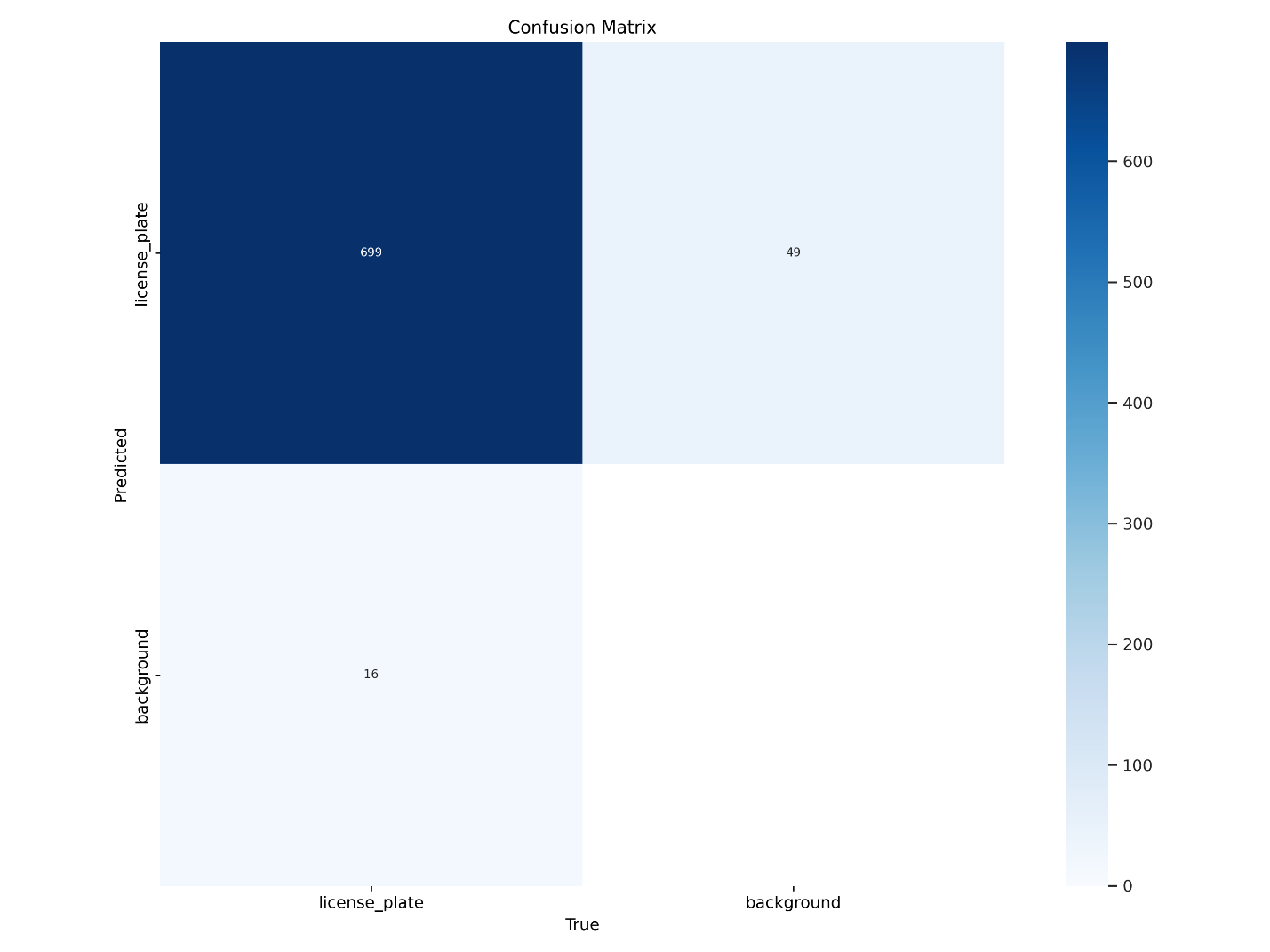
Các chỉ số cho thấy mô hình đạt độ chính xác cao trên tập test:

* **Precision (P)**: 96.9%, đảm bảo mô hình nhận diện đúng các biển số xe mà không dự đoán dư thừa.
* **Recall (R)**: 96.1%, thể hiện khả năng tìm kiếm hầu hết các biển số xe thực tế trong tập test.
* **mAP@50**: 99.0%, chỉ số đánh giá chính xác của mô hình ở mức IoU=0.5.
* **mAP@50-95**: 86.6%, đo độ chính xác trung bình trên nhiều ngưỡng IoU (từ 0.5 đến 0.95), phản ánh tính ổn định và hiệu quả cao của mô hình trên nhiều ngưỡng khác nhau.

**Phân Tích Confusion Matrix**

Ma trận nhầm lẫn được minh họa trong Hình 2 cung cấp cái nhìn sâu hơn về khả năng phân loại của mô hình. Các kết quả chính bao gồm:

* Đối với biển số xe (License Plate): Mô hình xác định chính xác 699 trường hợp và chỉ dự đoán nhầm 49 trường hợp.
* Đối với nền (Background): Mô hình đạt hiệu suất cao với chỉ 16 trường hợp sai sót.

****

Hình 21: Confusion Matrix trên tập kiểm tra

3.5 Xây dựng ứng dụng

Ứng dụng được phát triển trong dự án này tập trung vào **phát hiện phương tiện** và **nhận dạng biển số xe**, bao gồm hai thành phần chính là phát hiện và theo dõi. Dưới đây là mô tả chi tiết về cách tiếp cận và các công nghệ được sử dụng trong hệ thống.

3.5.1 Cải tiến EasyOCR lên PaddleOCR

Để cải thiện quá trình nhận diện biển số xe và các văn bản khác trong bối cảnh so sánh giữa EasyOCR và PaddleOCR, Chúng ta sẽ phân tích dựa trên các điểm đã được mô tả:

PaddleOCR vượt trội hơn EasyOCR về độ chính xác:

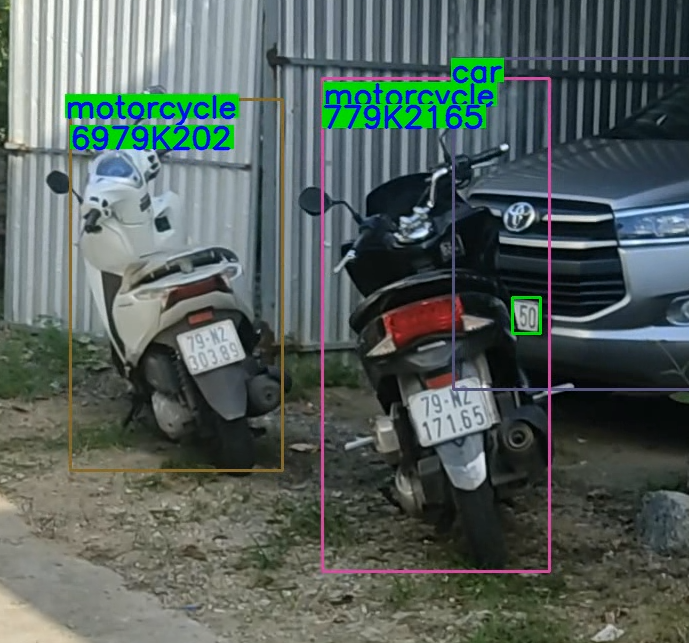
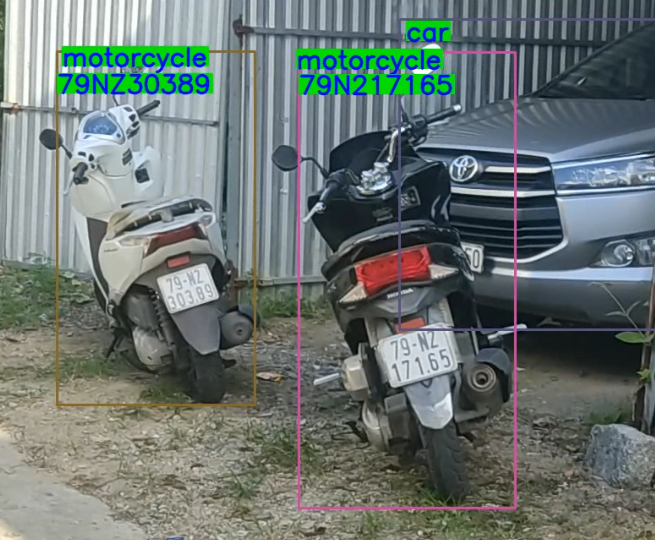
* Trong hình đầu tiên, PaddleOCR đạt mức chính xác cao hơn EasyOCR trên hầu hết các mục tiêu nhận diện. Ví dụ:
  + Từ “VICEROY” được nhận diện đúng với độ chính xác 97.1% so với chỉ 2.5% của EasyOCR.
  + Tương tự, “Yucatan Honey” được nhận diện chính xác với 93.0% ở PaddleOCR, so với chỉ 37.1% ở EasyOCR.
* Với các yếu tố phức tạp như dấu ngoặc hay văn bản bị méo, PaddleOCR vẫn duy trì độ chính xác cao hơn.



Hình 22 Nhận diện chữ của EasyOCR và PaddleOCR

**Độ chính xác trong nhận diện biển số xe**:

* PaddleOCR xác định biển số xe một cách chuẩn xác hơn so với EasyOCR, đặc biệt trong các trường hợp biển số bị che khuất hoặc nghiêng góc. Hình thứ hai và thứ ba minh họa rằng PaddleOCR xử lý lỗi ký tự tốt hơn:
  + Ví dụ: Biển số xe "79N217165" và "779K2165" trong PaddleOCR có độ chính xác cao hơn so với các kết quả từ EasyOCR.



Hình 23 Kết quả từ PaddleOCR

Hình 24 Kết quả từ EasyOCR

3.5.2 Phát hiện và theo dõi phương tiện

Quá trình phát hiện và theo dõi phương tiện trong ứng dụng được chia thành hai bước chính:

* **Phát hiện phương tiện:** Để phát hiện phương tiện, mô hình YOLOv8 được sử dụng. Đây là một mô hình tiên tiến đã được huấn luyện trên tập dữ liệu tùy chỉnh nhằm đạt độ chính xác cao trong việc phát hiện các loại phương tiện như xe máy, ô tô và xe tải.
* **Theo dõi phương tiện:** Sau khi phát hiện, thuật toán DeepSORT được áp dụng để theo dõi các phương tiện. DeepSORT là một bộ theo dõi mạnh mẽ, sử dụng kết hợp đặc điểm về chuyển động và ngoại hình để liên kết các phương tiện được phát hiện qua các khung hình. Điều này đảm bảo mỗi phương tiện được gán một ID duy nhất và được theo dõi liên tục trong suốt hành trình của nó.

3.5.3 Nhận dạng Biển Số Xe

Mô-đun nhận dạng biển số xe được thiết kế để trích xuất và phân tích thông tin chữ và số từ biển số xe. Quy trình gồm các bước sau:

* **Phát hiện biển số xe:**

Mô hình YOLOv8 cũng được sử dụng để phát hiện biển số xe. Mô hình này được tùy chỉnh để xác định vị trí biển số trên các phương tiện đã được phát hiện.

* **Nhận dạng ký tự quang học (OCR):**

Sau khi biển số được phát hiện, framework Baidu PaddleOCR được sử dụng để nhận diện ký tự. PaddleOCR là một công cụ OCR dựa trên học sâu, hoạt động hiệu quả và có độ chính xác cao. Nó xử lý vùng biển số đã được cắt ra để nhận diện và xuất ra các ký tự chữ và số trên biển.

**\* Quy trình hoạt động**

**Bước 1: Phát hiện phương tiện**

Phương tiện được phát hiện bằng mô hình YOLOv8 đã được training sẵn, tạo các khung bao quanh từng phương tiện.

**Bước 2: Theo dõi**

Các phương tiện được phát hiện sẽ được chuyển qua bộ theo dõi DeepSORT để gán ID duy nhất và duy trì danh tính của chúng qua các khung hình video.

**Bước 3: Phát hiện biển số**

Với mỗi phương tiện đã được theo dõi, biển số sẽ được xác định bằng mô hình YOLOv8.

**Bước 4: Xử lý OCR**

Vùng biển số được cắt ra và xử lý bằng PaddleOCR để trích xuất các ký tự trên biển.

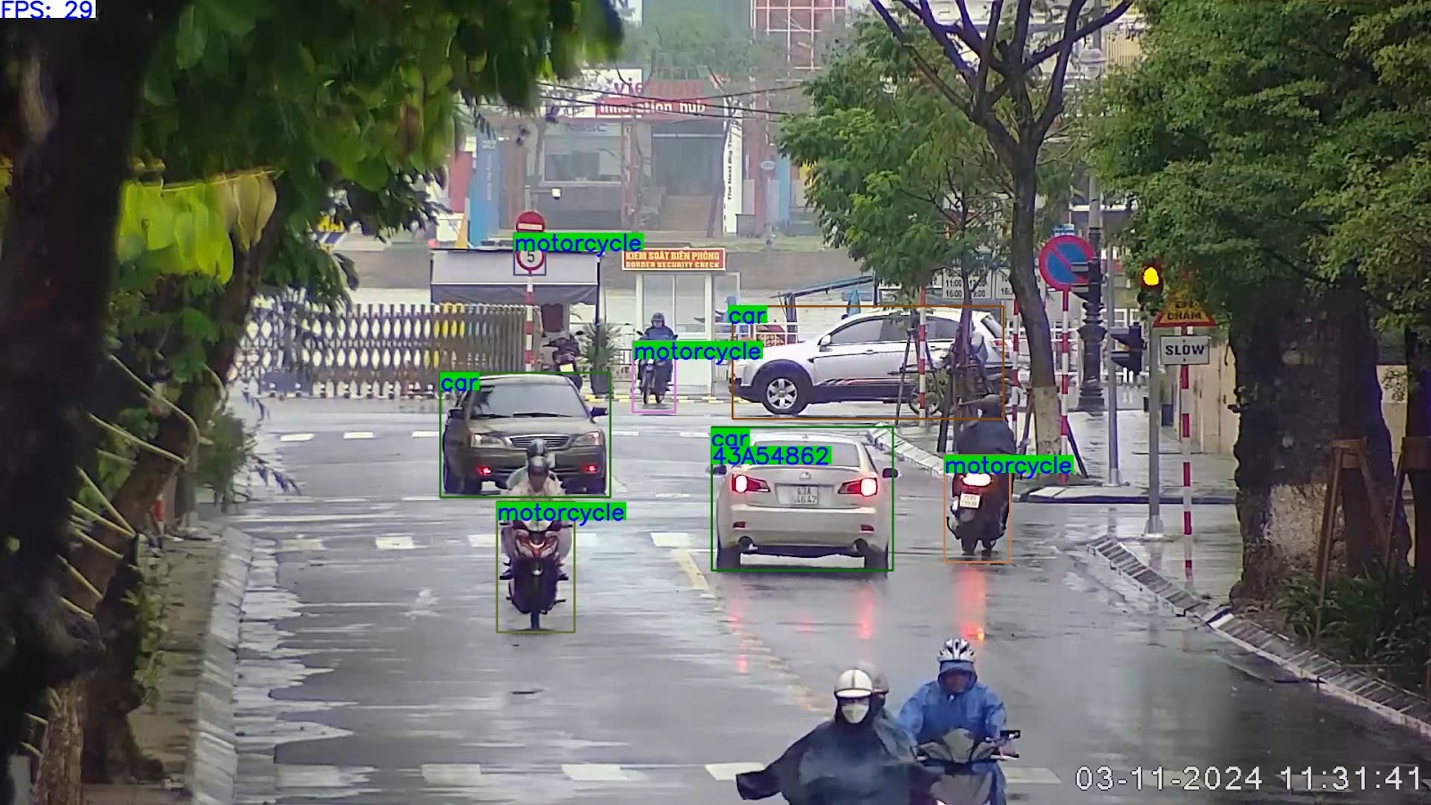
**Bước 5: Kết quả đầu ra**

Hệ thống xuất thông tin biển số xe đã nhận diện kèm với tên phương tiện được theo dõi, giúp việc giám sát và phân tích trở nên hiệu quả.

**\* Kết quả chạy từ code:**



Hình 25: Kết quả nhận diện biển số xe từ video chỉ với 1 xe



Hình 26: Kết quả nhận diện biển số xe từ camera ngã tư



Hình 27: Kết quả nhận diện biển số xe từ video ngã tư khác

3.6 Ứng dụng Android

Để kiểm thử khả năng nhận diện biển số xe trên thiết bị di động, một ứng dụng Android đơn giản đã được phát triển. Ứng dụng này tập trung vào việc phát hiện biển số xe và nhận dạng ký tự trên đó. Đây là phần mở rộng nhằm thử nghiệm tính năng trên nền tảng di động, không bao gồm các chức năng phát hiện phương tiện hoặc theo dõi.

**Quy trình Hoạt Động**

**Bước 1: Phát hiện biển số xe**

Mô hình YOLOv8, được chuyển đổi sang định dạng TensorFlow Lite (TFLite), được sử dụng để phát hiện vùng biển số trong hình ảnh hoặc video.

**Bước 2: Nhận dạng ký tự (OCR)**

Sau khi biển số được phát hiện, tính năng OCR tích hợp sẵn của Google ML Kit được sử dụng để trích xuất ký tự từ vùng biển số. Đây là giải pháp nhẹ và hiệu quả để thử nghiệm trên Android.

**Bước 3: Hiển thị kết quả**

Ứng dụng hiển thị ký tự đã nhận dạng trực tiếp trên màn hình cùng thời gian xử lý.

**Mục tiêu:** Ứng dụng được thiết kế để kiểm tra hiệu năng của mô hình trên nền tảng Android và tính khả thi của việc triển khai chức năng nhận dạng biển số trong môi trường tài nguyên hạn chế.

**Kết luận:** Ứng dụng Android mang tính thử nghiệm này cung cấp một cách tiếp cận đơn giản để đánh giá khả năng nhận diện biển số xe bằng OCR trên thiết bị di động, phục vụ mục tiêu kiểm thử và không hướng đến sử dụng trong các kịch bản phức tạp.



Hình 28: Phát hiện biển số xe trên điện thoại

Chương 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1 Kết luận

Trong đề tài này, chúng em đã thành công trong việc áp dụng Học sâu và các biến thể của mô hình YOLO (You Only Look Once) để phát triển công cụ nhận diện biển số xe. Chúng em đã thu thập và xử lý dữ liệu hình ảnh để tạo ra một bộ dữ liệu mẫu phục vụ cho việc huấn luyện mô hình. Đồng thời, môi trường thực thi thuật toán đã được thiết lập, mô hình YOLOv8 được xây dựng và huấn luyện để nhận diện biển số xe từ hình ảnh đầu vào.

Một ứng dụng đã được phát triển với khả năng phát hiện biển số và nhận diện ký tự số trên biển số xe. Giao diện người dùng đơn giản và dễ sử dụng, giúp các hệ thống như kiểm soát giao thông, quản lý bãi đỗ xe và giám sát an ninh tự động hoá quá trình nhận diện biển số xe.

Tuy nhiên, vẫn còn một số hạn chế cần được cải thiện. Cụ thể, khi tải nhiều mô hình cùng lúc và phát hiện nhiều xe và biển số xe trong một khung hình, tốc độ xử lý hiện tại khá thấp, ảnh hưởng đến khả năng thực thi trong môi trường thực tế. Ngoài ra, tỷ lệ chính xác trong một số trường hợp, đặc biệt khi biển số bị mờ hoặc không rõ ràng, cần được cải thiện.

4.2 Hướng phát triển

Trong tương lai, cần cải thiện tốc độ của mô hình để xử lý nhanh chóng và hiệu quả khi phải phát hiện nhiều biển số xe cùng lúc. Việc tối ưu hoá mô hình và giảm độ trễ trong quá trình phát hiện đối tượng là yếu tố then chốt để mô hình có thể áp dụng vào các hệ thống camera giám sát ngoài đường và nhận diện biển số xe trong thời gian thực.

Bên cạnh đó, cần tiếp tục nâng cao tỷ lệ chính xác của mô hình, đặc biệt là trong các trường hợp biển số xe bị mờ hoặc không rõ ràng. Việc phát triển thêm các tính năng như nhận diện biển số bị che khuất hoặc bị hư hại sẽ là hướng đi tiềm năng.

Ngoài ra, việc nghiên cứu và tích hợp các kỹ thuật khác như nhận diện biển số trong điều kiện ánh sáng yếu hoặc khi có nhiều phương tiện xuất hiện trong cùng một khung hình sẽ giúp mô hình trở nên mạnh mẽ và thực tế hơn.

# Tài liệu tham khảo

|  |  |
| --- | --- |
| [1] | "PaddleOCR Official Documentation," [Online]. Available: https://paddlepaddle.github.io/PaddleOCR/main/en/index.html#pp-ocrv3-chinese-model. |
| [2] | "Research Article on PaddleOCR," [Online]. Available: https://www.nature.com/articles/s41598-024-65272-1. |
| [3] | "Leveraging YOLO Algorithm and PaddleOCR in Machine Learning Applications," [Online]. Available: https://www.ijraset.com/research-paper/leveraging-yolo-algorithm-and-paddleocr-in-machine-learning-applications. |
| [4] | "PaddleOCR: A Comprehensive End-to-End OCR Solution," [Online]. Available: https://2021.desosa.nl/projects/paddleocr/posts/paddleocr-e2/#p3. |
| [5] | "YOLOv8 Models Benchmarking," [Online]. Available: https://docs.ultralytics.com/vi/models/yolov8/#can-i-benchmark-yolov8-models-for-performance. |
| [6] | "Deep SORT: Simple Online and Realtime Tracking," [Online]. Available: https://github.com/ModelBunker/Deep-SORT-PyTorch?fbclid=IwZXh0bgNhZW0CMTAAAR2AD-y9a4r0Hnu5SOQrJIDz5X84QimDC3h3fP0G-4CLwgmEuQZqcVIYc2I\_aem\_ZQ1DmaGep3VFKCDn7qPU0g. |