**TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN MÔN HỌC**

**NHẬN DIỆN NGƯỜI TRONG TRUNG TÂM THƯƠNG MẠI**

**Giảng viên hướng dẫn: ThS. ĐẶNG NHƯ PHÚ**

**Sinh viên thực hiện: HUỲNH QUY BÌNH**

**MSSV: 2100009844**

**Khoá: 21**

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

TP. HCM, tháng 5 năm 2024

**TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN MÔN HỌC**

**NHẬN DIỆN NGƯỜI TRONG TRUNG TÂM THƯƠNG MẠI**

**Giảng viên hướng dẫn: ThS. ĐẶNG NHƯ PHÚ**

**Sinh viên thực hiện: HUỲNH QUY BÌNH**

**MSSV: 2100009844**

**Khoá: 21**

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

TP. HCM, tháng 5 năm 2024

# LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành nhất tới tất cả các thầy cô trong khoa Công nghệ thông tin đã đóng góp và hỗ trợ em trong suốt quá trình thực hiện tiểu luận chuyên đề chuyên sâu - trí tuệ nhân tạo 2.

Em muốn bày tỏ lòng biết ơn sâu sắc tới giảng viên hướng dẫn của em là thầy Đặng Như Phú, người đã dành thời gian và kiến thức của mình để hỗ trợ và chỉ dẫn em trong suốt khoảng thời gian thực hiện. Sự hướng dẫn và những lời khuyên quý báu của thầy đã giúp em hiểu sâu hơn về trí tuệ nhân tạo và là nguồn động viên lớn để hoàn thành đồ án cơ sở này.

Em rất tự hào về thành quả và công sức đã bỏ ra để thực hiện tiểu luận chuyên đề này.

Một lần nữa, xin chân thành cảm ơn đến thầy người đã đồng hành cùng em trong hành trình này.

Em xin chân thành cảm ơn!

# LỜI MỞ ĐẦU

Trong bối cảnh của một xã hội hiện đại, trung tâm thương mại đã trở thành một phần không thể thiếu trong cuộc sống hàng ngày của con người. Chúng không chỉ là nơi mua sắm mà còn là địa điểm giải trí, giao lưu xã hội và thực hiện nhiều hoạt động khác nhau. Sự phát triển không ngừng của công nghệ đã mang lại nhiều tiện ích vượt trội, đặc biệt là trong lĩnh vực an ninh và quản lý. Trong số các công nghệ tiên tiến, nhận diện người bằng hình ảnh là một trong những phương pháp nổi bật, mang lại nhiều lợi ích quan trọng cho các trung tâm thương mại.

Nhận diện người không chỉ giúp cải thiện an ninh mà còn hỗ trợ quản lý hiệu quả lượng khách hàng, phân tích hành vi tiêu dùng và nâng cao trải nghiệm mua sắm. Công nghệ này giúp phát hiện và ngăn chặn các hành vi gian lận, trộm cắp, đồng thời cung cấp dữ liệu quý giá để các trung tâm thương mại có thể tối ưu hóa chiến lược kinh doanh của mình.

Trong tiểu luận này, chúng em sẽ tìm hiểu về công nghệ nhận diện người trong trung tâm thương mại, bao gồm các phương pháp, ứng dụng thực tiễn và những thách thức đặt ra. Chúng em cũng sẽ phân tích các lợi ích và hạn chế của công nghệ này, từ đó đề xuất các giải pháp nhằm tối ưu hóa việc áp dụng nhận diện người trong bối cảnh thực tiễn.

Thông qua nghiên cứu này, chúng em hy vọng sẽ mang lại cái nhìn toàn diện và sâu sắc về vai trò của công nghệ nhận diện người trong việc nâng cao chất lượng dịch vụ và an ninh tại các trung tâm thương mại, góp phần vào sự phát triển bền vững và hiệu quả của ngành hiện đại.

Em xin cảm ơn!

BM-ChT-11

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH  **TRUNG TÂM KHẢO THÍ** | **KỲ THI KẾT THÚC HỌC PHẦN**  **HỌC KỲ II NĂM HỌC 2023 – 2024** |

**PHIẾU CHẤM THI TIỂU LUẬN/ĐỒ ÁN**

Môn thi: Chuyên đề chuyên sâu trí tuệ nhân tạo 2 Lớp học phần: 21DTH2B

Nhóm sinh viên thực hiện:

1. Nguyễn Hữu Thịnh Tham gia đóng góp: 100%

2. Nguyễn Văn Nghĩa Tham gia đóng góp: 100%

3. Huỳnh Quy Bình Tham gia đóng góp: 100%

4. Tham gia đóng góp:

5. Tham gia đóng góp:

6. Tham gia đóng góp:

7. Tham gia đóng góp:

8. Tham gia đóng góp:

Ngày thi: 27/05/2024 Phòng thi: L.601

Đề tài tiểu luận/báo cáo của sinh viên: Nhận diện người trong trung tâm thương mại

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 CẢM ƠN ii](#_Toc167653741)

[LỜI MỞ ĐẦU iii](#_Toc167653742)

[Mục lục v](#_Toc167653743)

[KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT vii](#_Toc167653744)

[CHƯƠNG 1 1](#_Toc167653745)

[Cơ sở lý thuyết 1](#_Toc167653746)

[1.1. Học sâu (Deep Learning) và học máy (Machine Learning) và thị giác máy tính 1](#_Toc167653747)

[1.1.1. Học sâu (Deep Learning) 1](#_Toc167653748)

[1.1.2. Học máy (Machine Learning) 2](#_Toc167653749)

[1.2. Nhận diện đối tượng 3](#_Toc167653750)

[1.2.1. Thị giác máy tính là gì? 3](#_Toc167653751)

[1.2.2. Nhân diện đối tượng là gì? 4](#_Toc167653752)

[1.2.2.1. Có bao nhiêu thuật toán trong việc nhận diện đối tượng? 5](#_Toc167653753)

[1.2.2.2. Ưu điểm và nhược điểm của nhận diện đối tượng là gì? 7](#_Toc167653754)

[1.3. YOLO là gì? 9](#_Toc167653755)

[1.3.1. Lịch sử của YOLO? 10](#_Toc167653756)

[1.3.2. Nguyên lý hoạt động của YOLO? 12](#_Toc167653757)

[1.3.3. YOLOv8? 13](#_Toc167653758)

[1.3.4. Ai là tác giả của YOLOv8 14](#_Toc167653759)

[1.3.5. Các tính năng chính của YOLOv8 14](#_Toc167653760)

[1.3.6. YOLOv8 có ưu điểm và nhược điểm gì? 15](#_Toc167653761)

[1.3.7. YOLOv8 so với các phiên bản khác? 16](#_Toc167653762)

[1.4. Ultralytics là gì? 17](#_Toc167653763)

[1.5. Tracking là gì? 18](#_Toc167653764)

[1.5.1. Phân loại Tracking? 18](#_Toc167653765)

[1.6. Ngôn ngữ lập trình, thư viện và công cụ hỗ trợ 20](#_Toc167653766)

[1.6.1. Python 20](#_Toc167653767)

[1.6.2. Ưu nhược điểm của Python 21](#_Toc167653768)

[1.6.3. Thư viện OpenCV 22](#_Toc167653769)

[1.6.4. Ưu nhược điểm của OpenCV 23](#_Toc167653770)

[1.6.5. Những khó khăn trong việc nhận dạng đối tượng? 23](#_Toc167653771)

[CHƯƠNG 2 25](#_Toc167653772)

[Phân tích yêu cầu 25](#_Toc167653773)

[2.1. Yêu cầu đề tài 25](#_Toc167653774)

[2.2. Phân tích yêu cầu bài toán 25](#_Toc167653775)

[2.3. Thách thức trong đề tài 26](#_Toc167653776)

[CHƯƠNG 3 27](#_Toc167653777)

[Xây dựng mô hình 27](#_Toc167653778)

[3.1. Quy trình thực nghiệm mô hình 27](#_Toc167653779)

[3.2. Các thư viện được sử dụng 28](#_Toc167653780)

[CHƯƠNG 4 29](#_Toc167653781)

[Thực nghiệm mô hình 29](#_Toc167653782)

[4.1. Cài đặt các phụ thuộc 29](#_Toc167653783)

[4.2. Tracker 33](#_Toc167653784)

[4.3. Video để nhận diện 35](#_Toc167653785)

[4.4. Đối tượng để huấn luyện 36](#_Toc167653786)

[CHƯƠNG 5 37](#_Toc167653787)

[Kết luận và hướng phát triển 37](#_Toc167653788)

[5.1. Kết quả đạt được 37](#_Toc167653789)

[5.2. Hạn chế của đề tài 38](#_Toc167653790)

[5.3. Hướng phát triển 39](#_Toc167653791)

[TÀI LIỆU THAM KHẢO 40](#_Toc167653792)

# KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT

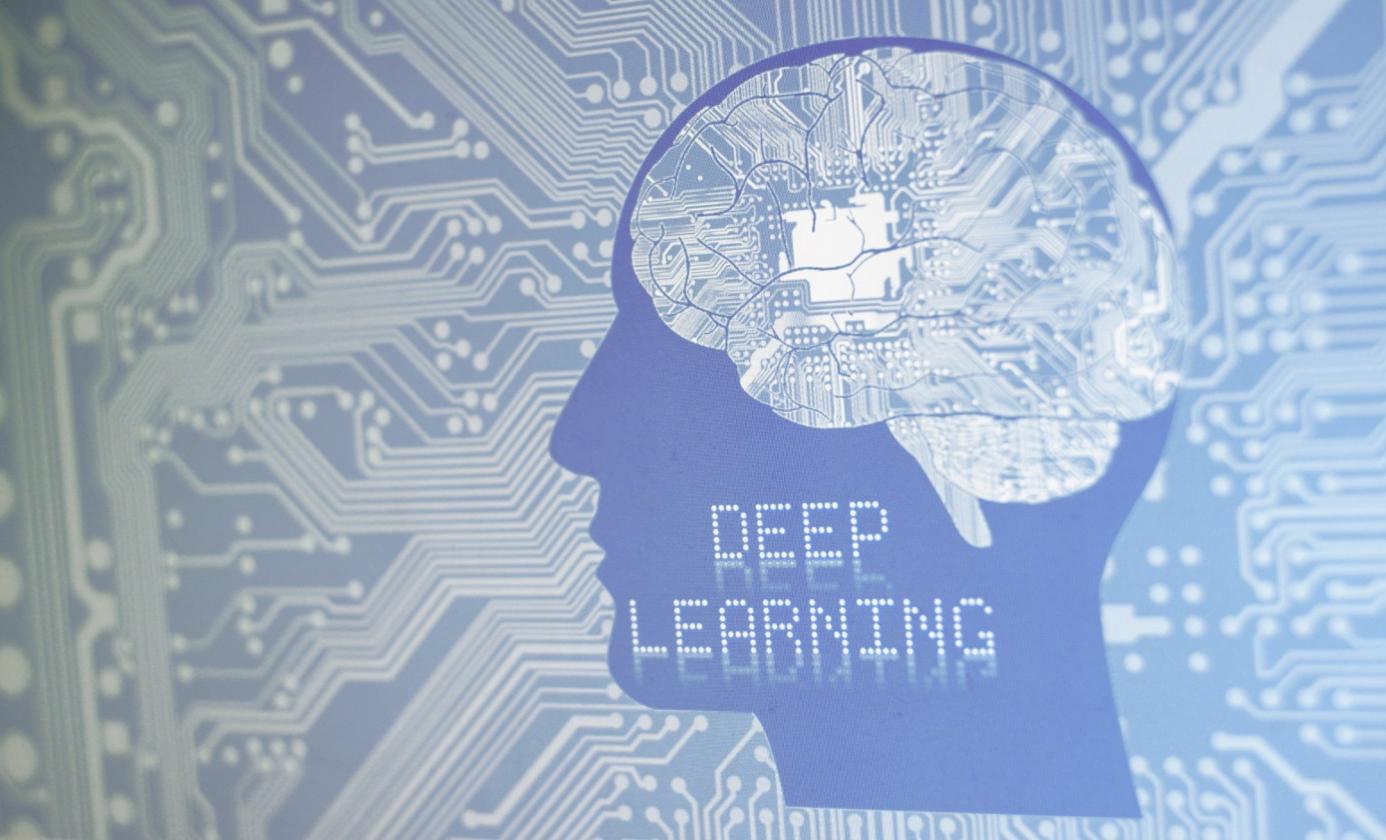
|  |  |
| --- | --- |
| **Chữ viết tắt** | **Ý nghĩa** |
| R-CNN | Region-based Convolutional Neural Networks |
| YOLO | You Only Look Once |
| SSD | Single Shot Multibox Detector |
| IBM | International Business Machines Corporation (Tập đoàn về công nghệ máy tính) |

# CHƯƠNG 1

# Cơ sở lý thuyết

## 1.1. Học sâu (Deep Learning) và học máy (Machine Learning) và thị giác máy tính

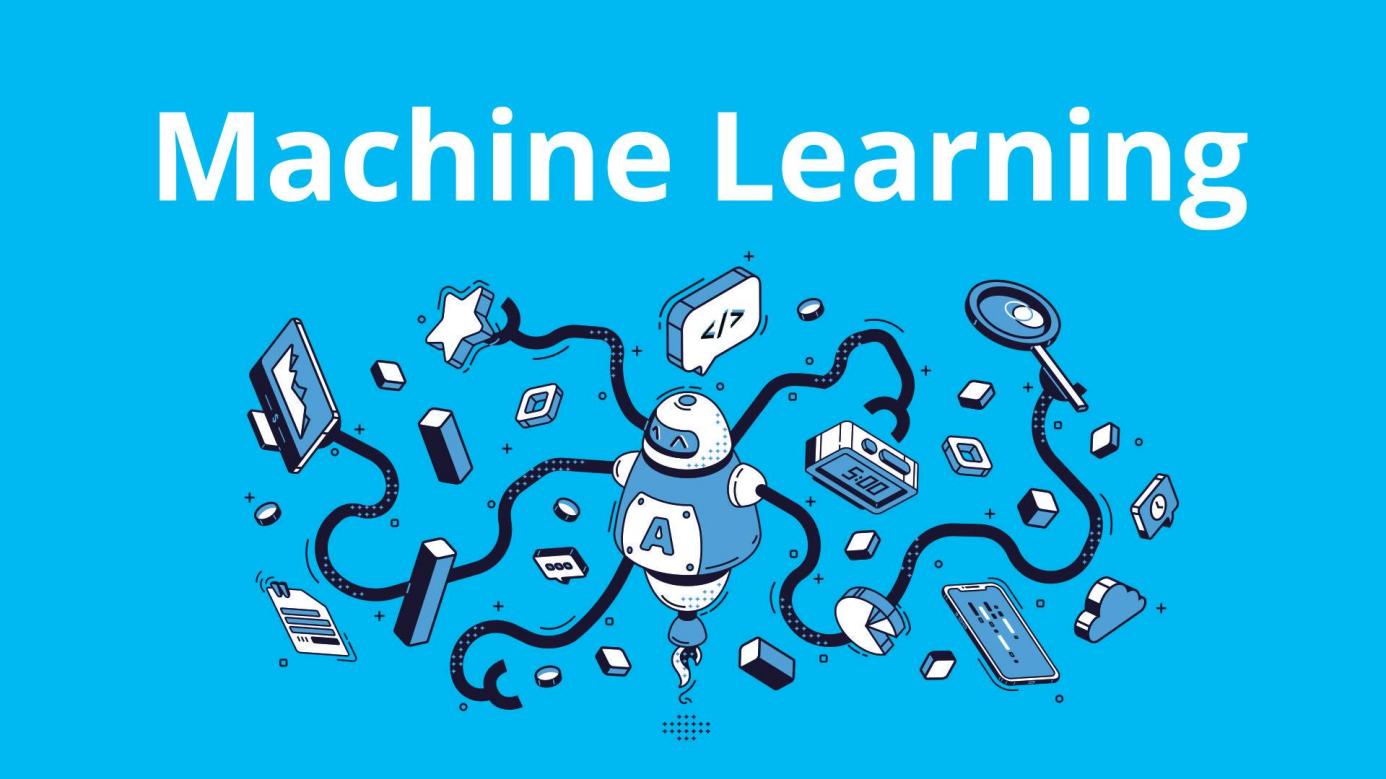
### 1.1.1. Học sâu (Deep Learning)

Học sâu, một phương pháp tiên tiến trong lĩnh vực trí tuệ nhân tạo (AI), được phát triển để giúp máy tính xử lý dữ liệu một cách tự động và thông minh, mô phỏng theo cách thức hoạt động của bộ não con người. Các mô hình học sâu, hay còn gọi là mạng nơ-ron sâu, có khả năng nhận diện và phân tích các mẫu phức tạp trong các dạng dữ liệu đa dạng như hình ảnh, văn bản, âm thanh và nhiều loại dữ liệu khác. Thông qua việc học từ lượng dữ liệu lớn, các mô hình này có thể tạo ra những thông tin chuyên sâu, đưa ra dự đoán chính xác và thậm chí thực hiện các tác vụ phức tạp một cách tự động.

Công nghệ học sâu không chỉ dừng lại ở việc nhận diện hình ảnh mà còn mở rộng ra nhiều ứng dụng khác như phân tích ngôn ngữ tự nhiên, nhận diện giọng nói, và dịch thuật tự động. Những hệ thống này có thể mô tả chi tiết hình ảnh, chuyển đổi giọng nói thành văn bản một cách chính xác, và thậm chí tạo ra văn bản mới từ những dữ liệu đầu vào. Nhờ vào khả năng học hỏi và cải thiện liên tục, học sâu đang dần thay thế con người trong nhiều nhiệm vụ đòi hỏi trí tuệ và sự phân tích phức tạp.

Các ứng dụng của học sâu không chỉ giúp tiết kiệm thời gian và công sức mà còn mở ra những khả năng mới trong nhiều lĩnh vực khác nhau như y tế, tài chính, giải trí, và dịch vụ. Trong y tế, học sâu giúp chẩn đoán bệnh từ hình ảnh y học với độ chính xác cao; trong tài chính, nó giúp dự đoán xu hướng thị trường và phát hiện gian lận. Ở các lĩnh vực như giải trí và dịch vụ, học sâu cũng đóng vai trò quan trọng trong việc nâng cao trải nghiệm người dùng thông qua các hệ thống gợi ý và tương tác tự động.

### 1.1.2. Học máy (Machine Learning)

Máy học là một lĩnh vực khoa học chuyên nghiên cứu và phát triển các thuật toán cùng mô hình thống kê, giúp hệ thống máy tính thực hiện các tác vụ thông qua việc nhận diện khuôn mẫu và suy luận mà không cần hướng dẫn cụ thể. Thay vì được lập trình với những chỉ dẫn chi tiết cho từng nhiệm vụ, các hệ thống máy tính sử dụng thuật toán máy học để xử lý một khối lượng lớn dữ liệu từ quá khứ, qua đó xác định các khuôn mẫu tiềm ẩn trong dữ liệu đó.

Quá trình này cho phép máy tính học hỏi và cải thiện khả năng dự đoán của mình. Khi đã được huấn luyện với một lượng lớn dữ liệu, các hệ thống máy học có thể áp dụng những hiểu biết đã học được để đưa ra dự đoán chính xác hơn dựa trên các tập dữ liệu đầu vào mới. Điều này có nghĩa là máy tính không chỉ nhận biết và xử lý thông tin một cách hiệu quả hơn mà còn tự động hóa được nhiều tác vụ phức tạp, từ nhận diện hình ảnh và giọng nói, đến dự đoán xu hướng trong tài chính và y tế.

Một trong những lợi thế lớn nhất của máy học là khả năng thích nghi và tự cải thiện. Khi lượng dữ liệu đầu vào tăng lên và thay đổi, các mô hình máy học có thể liên tục điều chỉnh để phản ánh những biến đổi này, từ đó cung cấp những dự đoán và phân tích chính xác hơn. Điều này đặc biệt hữu ích trong những lĩnh vực đòi hỏi sự linh hoạt và khả năng phản ứng nhanh chóng với những thay đổi, chẳng hạn như trong các ứng dụng thương mại điện tử, y học, hay quản lý rủi ro tài chính.

## 1.2. Nhận diện đối tượng

### 1.2.1. Thị giác máy tính là gì?

Thị giác máy tính là một lĩnh vực khoa học máy tính chuyên sâu, tập trung vào việc tái tạo các khả năng phức tạp của hệ thống thị giác con người. Mục tiêu của thị giác máy tính là phát triển các thuật toán và mô hình cho phép máy tính xác định, nhận diện và xử lý các đối tượng trong hình ảnh và video một cách tự động và chính xác, tương tự như cách con người làm.

Thông qua việc phân tích dữ liệu hình ảnh từ camera, video hoặc các nguồn tương tự, hệ thống thị giác máy tính có thể thực hiện nhiều tác vụ quan trọng như nhận diện khuôn mặt, phát hiện vật thể, phân đoạn hình ảnh, và theo dõi chuyển động. Các ứng dụng của thị giác máy tính rất đa dạng và phong phú, từ an ninh giám sát, y tế, ô tô tự lái, đến nhận diện sản phẩm trong các hệ thống bán lẻ tự động.

Một trong những thách thức lớn của thị giác máy tính là khả năng xử lý và hiểu được các hình ảnh trong các điều kiện ánh sáng, góc nhìn và chất lượng khác nhau. Để giải quyết những thách thức này, các nhà khoa học đã phát triển nhiều kỹ thuật tiên tiến như học sâu, mạng nơ-ron tích chập (Convolutional Neural Networks - CNNs), và các thuật toán tối ưu hóa, nhằm cải thiện độ chính xác và hiệu suất của các hệ thống thị giác máy tính.

Nhờ vào những tiến bộ trong lĩnh vực học sâu và xử lý dữ liệu lớn, các hệ thống thị giác máy tính ngày càng trở nên mạnh mẽ và linh hoạt hơn. Chúng không chỉ giúp tự động hóa nhiều tác vụ phức tạp mà còn mở ra những khả năng mới trong việc phân tích và hiểu biết về thế giới xung quanh thông qua hình ảnh.



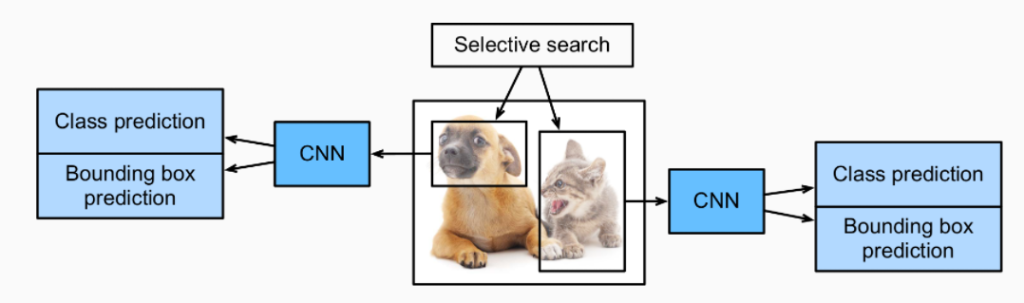
### 1.2.2. Nhân diện đối tượng là gì?



Nhận diện đối tượng (Object Detection) là một phần quan trọng trong lĩnh vực thị giác máy tính. Đây là quá trình sử dụng trí tuệ nhân tạo để xác định và phân loại các vật thể khác nhau trong hình ảnh hoặc video. Mục tiêu của nhận diện đối tượng là xác định vị trí và loại của các đối tượng trong một hình ảnh hoặc video. Đối tượng có thể là bất cứ thứ gì từ con người, động vật, đến vật thể và cả các phần của hình ảnh như khuôn mặt, xe hơi, đồ vật trong phòng, v.v.

#### 1.2.2.1. Có bao nhiêu thuật toán trong việc nhận diện đối tượng?

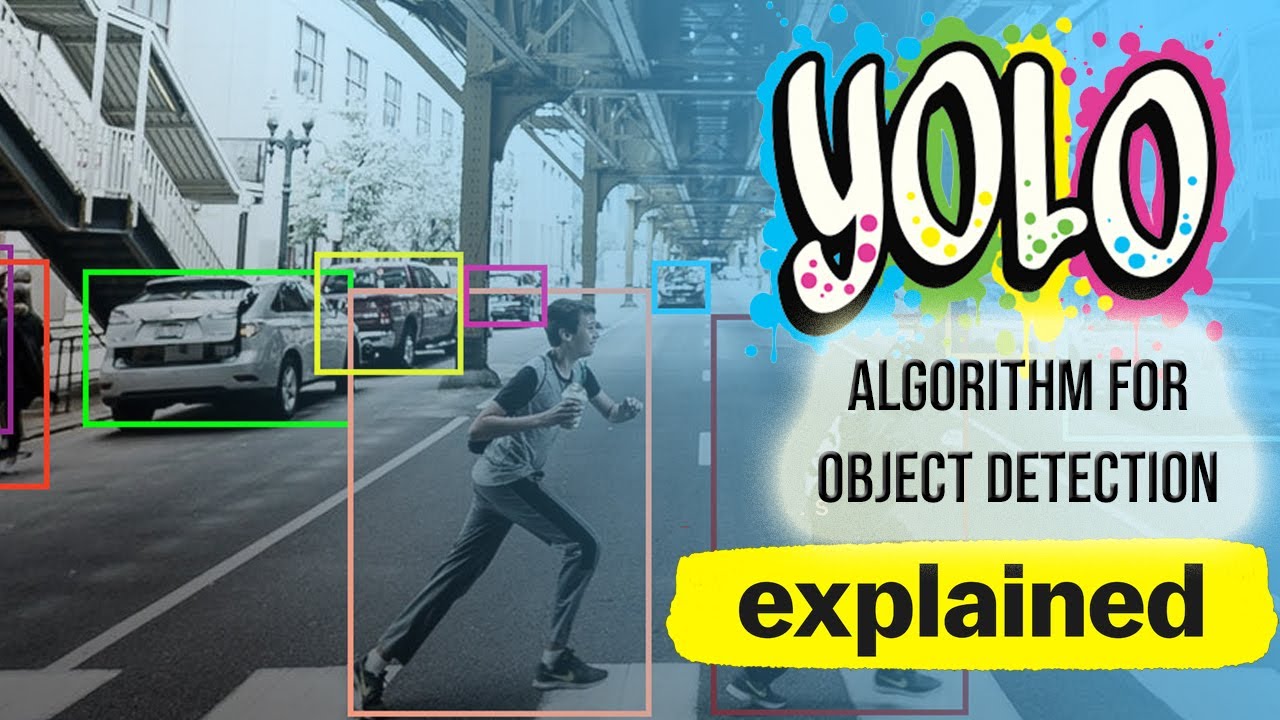
Có nhiều thuật toán và phương pháp khác nhau được sử dụng trong việc nhận diện đối tượng, mỗi thuật toán thường có ưu điểm và hạn chế riêng. Trong bài báo cáo này bọn em chỉ liệt kê những phương pháp chính trong nhận diện đối tượng.



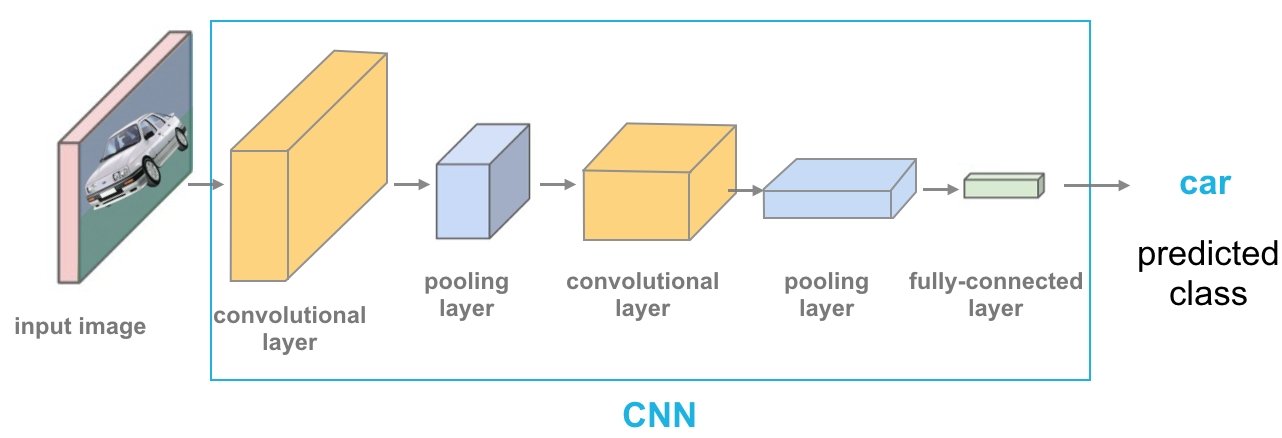
R-CNN (Region-based Convolutional Neural Networks): Bao gồm các biến thể như R-CNN, Fast R-CNN, và Faster R-CNN, các mô hình này sử dụng một mạng neural để đề xuất vùng ứng viên (region proposals) và sau đó phân loại các vùng này.



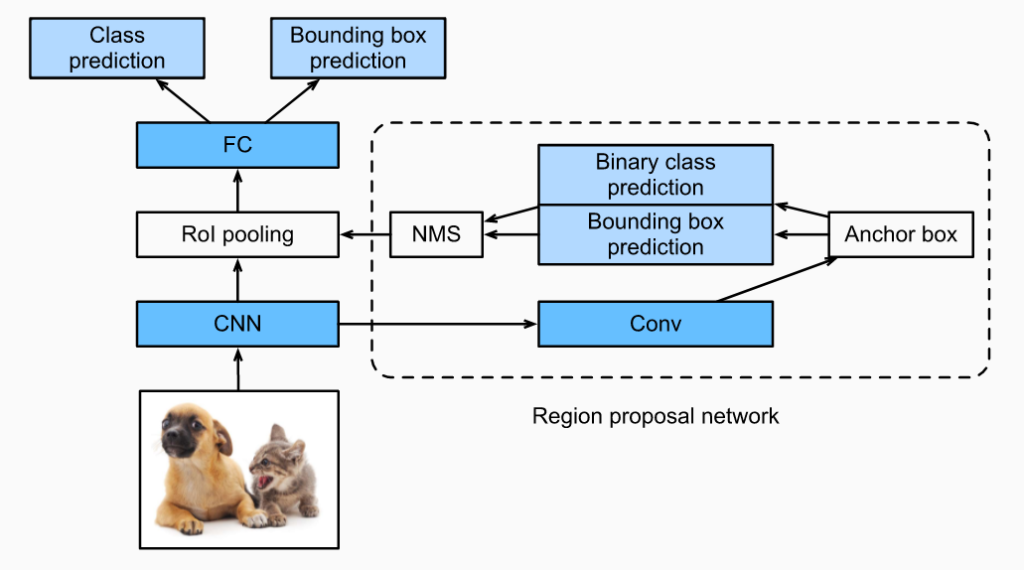
YOLO (You Only Look Once): YOLO là một phương pháp nhận diện đối tượng nhanh chóng và hiệu quả, hoạt động bằng cách dự đoán bounding boxes và các lớp đối tượng trực tiếp từ toàn bộ hình ảnh một lần.



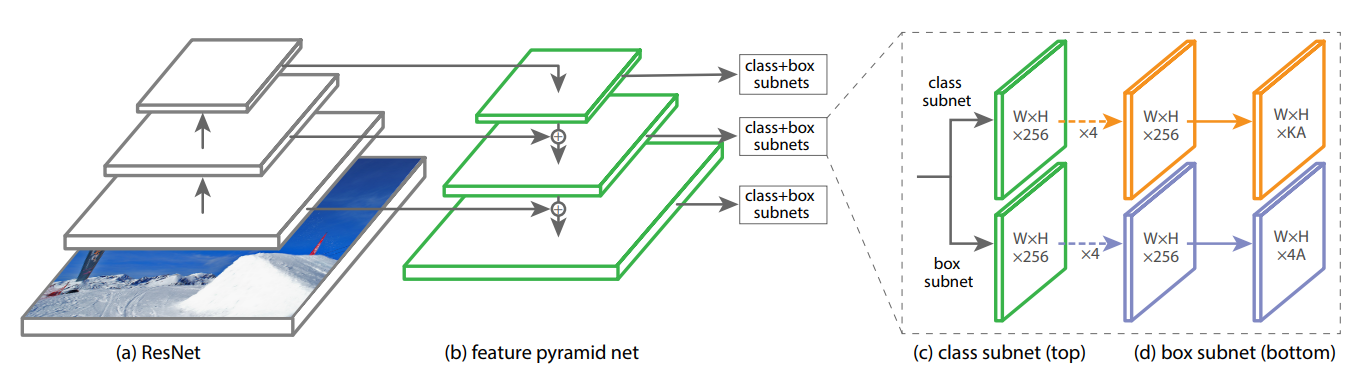
SSD (Single Shot Multibox Detector): SSD là một phương pháp nhận diện đối tượng khác, cũng như YOLO, nhưng nó sử dụng một số lượng nhỏ các lớp phát hiện ở nhiều tỷ lệ khác nhau để cải thiện độ chính xác.



Mask R-CNN: Một biến thể của R-CNN, Mask R-CNN không chỉ nhận diện đối tượng mà còn tạo ra mặt nạ cho mỗi đối tượng, cho phép phân loại đối tượng và segment hình ảnh cùng một lúc.



Faster R-CNN: Một biến thể nhanh hơn của R-CNN, Faster R-CNN sử dụng một mạng neural để đề xuất các vùng ứng viên một cách hiệu quả hơn.



RetinaNet: RetinaNet là một mô hình nhận diện đối tượng dựa trên mạng neural mạnh mẽ nhưng đơn giản, được thiết kế để giải quyết vấn đề đối với các đối tượng nhỏ và đối tượng hiếm.

#### 1.2.2.2. Ưu điểm và nhược điểm của nhận diện đối tượng là gì?

- Ưu điểm:

* Tự động hóa công việc: Nhận diện đối tượng giúp tự động hóa nhiều công việc, từ quản lý hàng hóa trong nhà kho đến theo dõi tình trạng giao thông.
* Cải thiện an ninh và an toàn: Trong lĩnh vực an ninh, việc nhận diện đối tượng có thể giúp phát hiện và ngăn chặn các hành vi đe dọa, cải thiện an ninh cho cộng đồng.
* Tăng cường trải nghiệm người dùng: Trong các ứng dụng di động và các dịch vụ trực tuyến, nhận diện đối tượng có thể cung cấp trải nghiệm người dùng tốt hơn thông qua các tính năng như nhận dạng khuôn mặt hoặc tìm kiếm hình ảnh.
* Nâng cao hiệu suất sản xuất: Trong ngành công nghiệp, nhận diện đối tượng có thể giúp cải thiện quy trình sản xuất và kiểm soát chất lượng, giảm thiểu lãng phí và tăng năng suất.

- Nhược điểm:

* Độ chính xác không đảm bảo: Các hệ thống nhận diện đối tượng có thể không luôn hoạt động chính xác 100%, đặc biệt là đối với các điều kiện ánh sáng kém, góc chụp không lý tưởng hoặc đối tượng mờ.
* Tiêu tốn tài nguyên tính toán: Một số phương pháp nhận diện đối tượng đòi hỏi nhiều tài nguyên tính toán, đặc biệt là trong các ứng dụng thời gian thực hoặc trên thiết bị có tài nguyên hạn chế.
* Cần dữ liệu huấn luyện lớn: Để đạt được hiệu suất tốt, các mô hình nhận diện đối tượng thường cần được huấn luyện trên các tập dữ liệu lớn và đa dạng, điều này có thể tốn kém và phức tạp.
* Đối với đối tượng mới: Trong một số trường hợp, việc nhận diện đối tượng của các đối tượng mới, chưa từng xuất hiện trong dữ liệu huấn luyện, có thể gặp khó khăn và cần phải được cập nhật thường xuyên.

## 1.3. YOLO là gì?



YOLO, viết tắt của "You Only Look Once", là một phương pháp tiên tiến trong lĩnh vực nhận diện đối tượng và phân loại ảnh. YOLO là một mạng neural tích chập (Convolutional Neural Network - CNN) được thiết kế để nhận diện đồng thời nhiều đối tượng trong một hình ảnh và dự đoán vị trí cũng như lớp của chúng một cách nhanh chóng và chính xác.

Điểm đặc biệt của YOLO so với các phương pháp truyền thống khác là nó thực hiện việc nhận diện đối tượng chỉ trong một lần chạy mạng neural trên toàn bộ bức ảnh. Các phương pháp truyền thống thường chia quá trình nhận diện thành nhiều bước, như đề xuất vùng và sau đó phân loại từng vùng, dẫn đến tốc độ xử lý chậm hơn. Trong khi đó, YOLO xử lý toàn bộ ảnh chỉ trong một bước duy nhất, cho phép hệ thống xác định vị trí và nhận diện đối tượng một cách đồng thời. Điều này giúp YOLO trở nên cực kỳ nhanh chóng và hiệu quả, đặc biệt hữu ích trong các ứng dụng thời gian thực như xe tự lái, giám sát an ninh, và nhận diện đối tượng trong video.

Nhiều phiên bản của YOLO đã được phát triển để cải thiện hiệu suất và độ chính xác.

Các phiên bản này bao gồm:

- YOLOv1: Phiên bản đầu tiên, giới thiệu khái niệm xử lý toàn bộ hình ảnh trong một lần chạy.

- YOLOv2 (còn gọi là YOLO9000): Cải tiến về mặt tốc độ và độ chính xác, bổ sung khả năng nhận diện nhiều hơn 9000 loại đối tượng khác nhau.

- YOLOv3: Tăng cường độ chính xác bằng cách sử dụng kiến trúc mạng sâu hơn và phức tạp hơn, hỗ trợ việc phát hiện đối tượng ở nhiều cấp độ khác nhau trong hình ảnh.

- YOLOv4: Tập trung vào việc tối ưu hóa hiệu suất, cải thiện tốc độ và độ chính xác, và bổ sung các kỹ thuật hiện đại như mô hình học sâu được tối ưu hóa và các phương pháp học đa nhiệm (multi-task learning).

Nhờ vào những cải tiến này, YOLO ngày càng trở nên mạnh mẽ và linh hoạt, đáp ứng được nhu cầu của nhiều ứng dụng thực tiễn khác nhau. Khả năng xử lý nhanh chóng và chính xác của YOLO đã làm cho nó trở thành một công cụ quan trọng trong nhiều lĩnh vực, từ công nghệ xe tự lái, giám sát an ninh, đến phân tích video và nhiều ứng dụng khác.

### 1.3.1. Lịch sử của YOLO?



YOLO (You Only Look Once) là một mô hình phát hiện đối tượng và phân đoạn hình ảnh phổ biến, được phát triển bởi Joseph Redmon và Ali Farhadi tại Đại học Washington. Ra mắt lần đầu vào năm 2015, YOLO nhanh chóng trở nên phổ biến nhờ vào tốc độ và độ chính xác cao trong việc nhận diện đối tượng trong hình ảnh và video. Dưới đây là sự phát triển và cải tiến của các phiên bản YOLO qua các năm:

- YOLOv1 (2015): Phiên bản đầu tiên, giới thiệu phương pháp xử lý toàn bộ hình ảnh trong một lần chạy, mang lại sự nhanh chóng và hiệu quả vượt trội so với các phương pháp truyền thống.

- YOLOv2 (2016): Cải tiến mô hình ban đầu bằng cách kết hợp chuẩn hóa hàng loạt (batch normalization), sử dụng hộp neo (anchor boxes) và cụm kích thước (dimension clustering). Những thay đổi này giúp cải thiện độ chính xác và hiệu suất của mô hình.

- YOLOv3 (2018): Nâng cao hiệu suất của mô hình bằng cách sử dụng mạng đường trục (backbone network) hiệu quả hơn, nhiều hộp neo và gộp kim tự tháp không gian (spatial pyramid pooling). Những cải tiến này giúp mô hình nhận diện đối tượng ở nhiều cấp độ khác nhau trong hình ảnh.

- YOLOv4 (2020): Giới thiệu những cải tiến như tăng cường dữ liệu Mosaic, đầu phát hiện không neo mới và chức năng mất mát mới. Những cải tiến này giúp tăng cường độ chính xác và tốc độ xử lý của mô hình.

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

- YOLOv6 (2022): Được Meituan mở nguồn mở và sử dụng trong nhiều robot giao hàng tự động của công ty, cải thiện khả năng xử lý và ứng dụng thực tiễn.

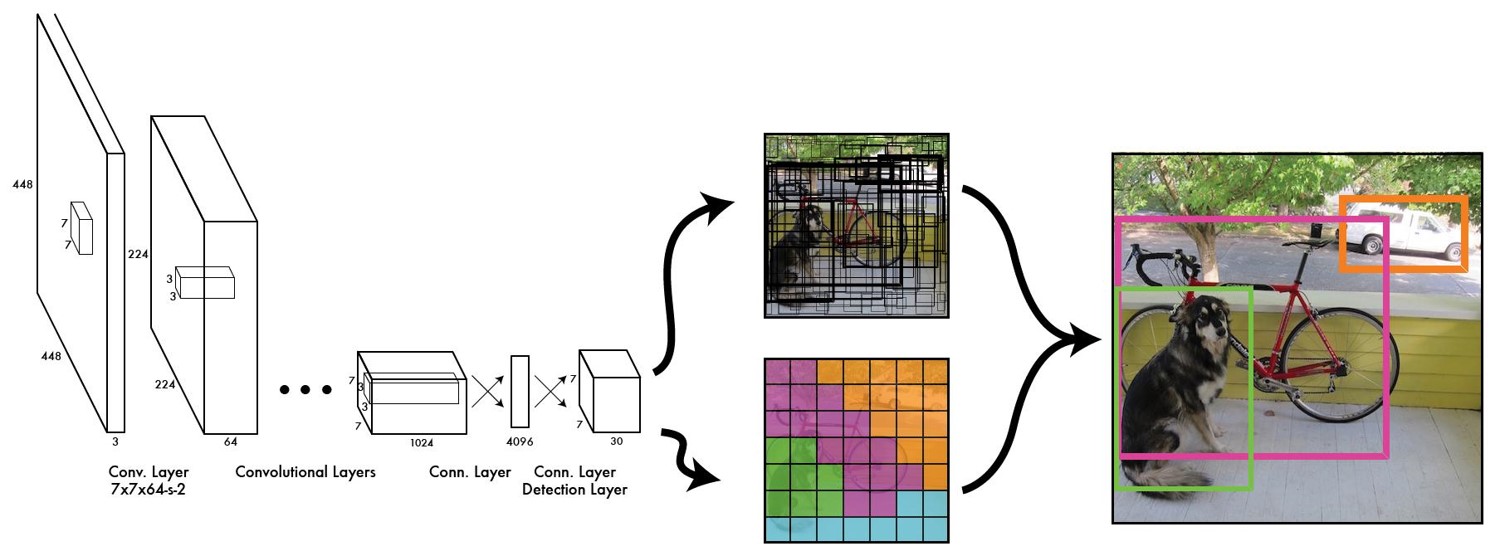
- YOLOv7: Thêm các tác vụ bổ sung như ước tính tư thế trên bộ dữ liệu điểm chính COCO, mở rộng khả năng ứng dụng của mô hình trong các lĩnh vực khác nhau.

- YOLOv8: Phiên bản mới nhất của YOLO được phát triển bởi Ultralytics. YOLOv8 được xây dựng dựa trên sự thành công của các phiên bản trước, giới thiệu các tính năng và cải tiến mới để nâng cao hiệu suất, tính linh hoạt và hiệu quả. YOLOv8 hỗ trợ đầy đủ các tác vụ AI thị giác, bao gồm phát hiện, phân đoạn, ước tính tư thế, theo dõi và phân loại, mang lại sự linh hoạt cao cho người dùng.

- YOLOv9: Giới thiệu các phương pháp sáng tạo như Programmable Gradient Information (PGI) và Generalized Efficient Layer Aggregation Network (GELAN), nhằm nâng cao hơn nữa hiệu suất và khả năng của mô hình trong việc xử lý hình ảnh và video.

Mỗi phiên bản YOLO đều mang lại những cải tiến quan trọng, giúp nâng cao khả năng nhận diện đối tượng và phân đoạn hình ảnh, đồng thời mở rộng phạm vi ứng dụng trong các lĩnh vực khác nhau từ giám sát an ninh, xe tự lái đến các hệ thống tự động hóa và robot.

### 1.3.2. Nguyên lý hoạt động của YOLO?

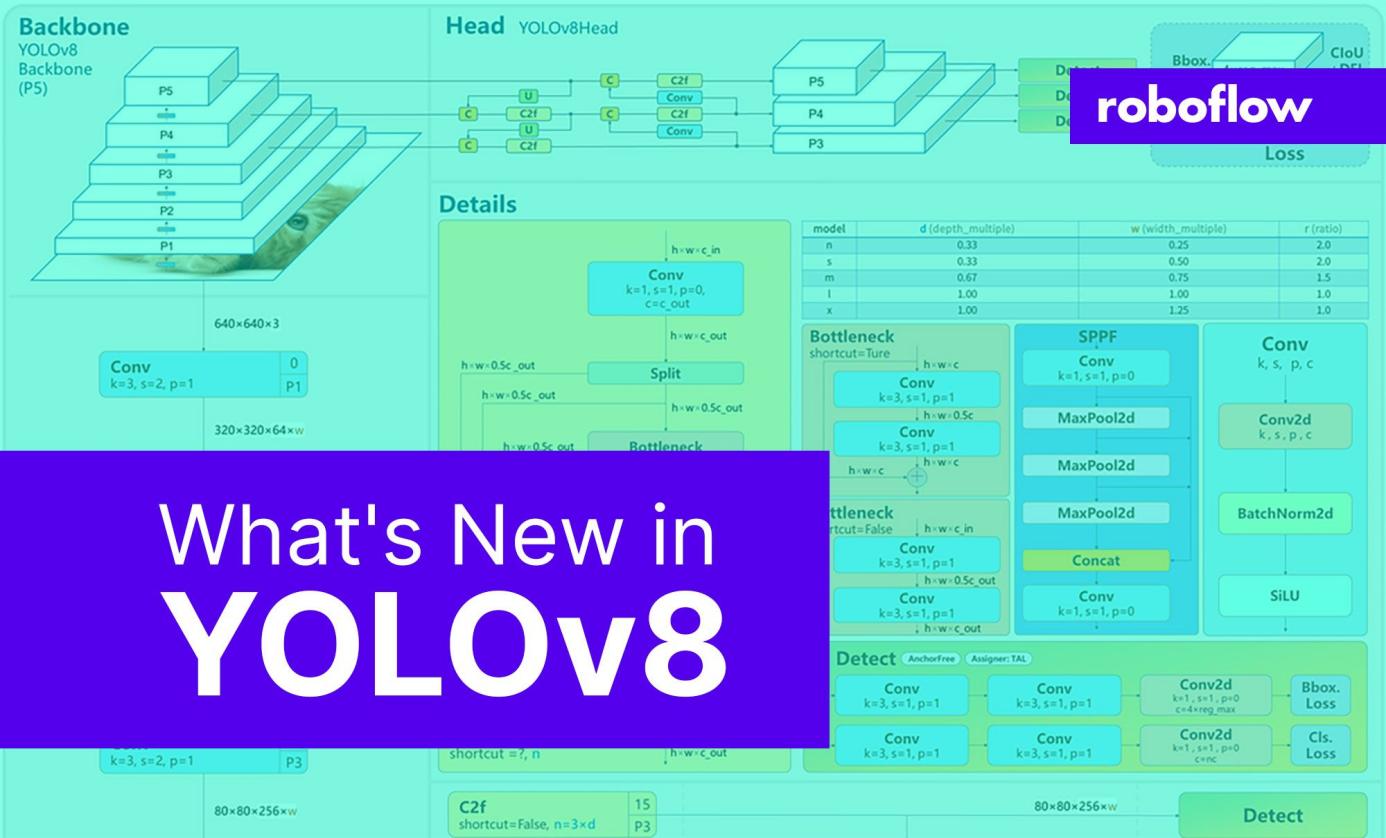


Đầu vào của mô hình là một ảnh, mô hình sẽ nhận dạng ảnh đó có đối tượng nào hay không, sau đó sẽ xác định tọa độ của đối tượng trong bức ảnh. ẢNh đầu vào được chia thành thành *S*×*S* ô thường thì sẽ là 3×33×3, 7×77×7, 9×99×9... việc chia ô này có ảnh hưởng tới việc mô hình phát hiện đối tượng. Với Input là 1 ảnh, đầu ra mô hình là một ma trận 3 chiều có kích thước *S*×*S*×(5×*N*+*M*) với số lượng tham số mỗi ô là (5×*N*+*M*) với N và M lần lượt là số lượng Box và Class mà mỗi ô cần dự đoán. Ví dụ với hình ảnh trên chia thành 7×7 ô, mỗi ô cần dự đóan 2 bounding box và 3 object: con chó, ô tô, xe đạp thì output là 7×7×13, mỗi ô sẽ có 13 tham số, kết quả trả về (7×7×2=98) bounding box. Chúng ta sẽ cùng giải thích con số (5×*N*+*M*) được tính như thế nào.

Dự đoán mỗi bounding box gồm 5 thành phần : (x, y, w, h, prediction) với (x, y ) là tọa độ tâm của bounding box, (w, h) lần lượt là chiều rộng và chiều cao của bounding box, prediction được định nghĩa Pr(*Object*)∗ *IOU*(*pred*,*truth*) . Với hình ảnh trên như ta tính mỗi ô sẽ có 13 tham số, ta có thể hiểu đơn giản như sau tham số thứ 1 sẽ chỉ ra ô đó có chứa đối tượng nào hay không P(Object), tham số 2, 3, 4, 5 sẽ trả về x, y, w, h của Box1. Tham số 6, 7, 8, 9, 10 tương tự sẽ Box2, tham số 11, 12, 13 lần lượt là xác suất ô đó có chứa object1(P(chó|object), object2(P(ô tô|object)), object3(P( xe đạp|object)).

Lưu ý rằng tâm của bounding box nằm ở ô nào thì ô đó sẽ chứa đối tượng, cho dù đối tượng có thể ở các ô khác thì cũng sẽ trả về là 0. Vì vậy việc mà 1 ô chứa 2 hay nhiều tâm của bouding box hay đối tượng thì sẽ không thể detect được, đó là một hạn chế của mô hình YOLO1, vậy ta cần phải tăng số lượng ô chia trong 1 ảnh lên đó là lí do vì sao mình nói việc chia ô có thể làm ảnh hưởng tới việc mô hình phát hiện đối tượng.

### 1.3.3. YOLOv8?



YOLOv8 là phiên bản mới nhất trong dòng mô hình YOLO. YOLO là viết tắt của cụm từ “You Only Look Once”, tức chỉ khả năng dự đoán tất cả các đối tượng có mặt trong một hình ảnh chỉ trong một lần truyền dữ liệu.

Điểm khác biệt chính mà YOLO mang lại là cách đặt vấn đề. Tác giả đã đặt lại vấn đề phát hiện đối tượng dưới dạng một bài toán hồi quy (dự đoán tọa độ hộp giới hạn) thay vì phân loại.

Các mô hình YOLO được huấn luyện trước trên các bộ dữ liệu lớn như COCO và ImageNet. Điều này cho phép chúng vừa có khả năng cung cấp dự đoán cực kỳ chính xác với các lớp đã được huấn luyện, vừa có thể học các lớp mới một cách tương đối dễ dàng.

Các mô hình YOLO cũng có tốc độ huấn luyện nhanh hơn, với độ chính xác cao và kích thước mô hình nhỏ. Chúng có thể được huấn luyện trên các GPU đơn, do đó dễ tiếp cận hơn đối với các nhà phát triển.

Được công bố vào đầu năm 2023, YOLOv8 đã mang lại nhiều điểm tích cực so với phiên bản tiền nhiệm, như phát hiện không dùng anchor, giới thiệu lớp tích chập C3 và tăng cường mosaic.

### 1.3.4. Ai là tác giả của YOLOv8

YOLOv8 được viết và duy trì bởi nhóm Ultralytics. Các mô hình YOLO ban đầu do Joseph Redmon, một nhà khoa học máy tính, tạo ra. Ông đã phát triển ba phiên bản YOLO, với phiên bản thứ ba là YOLOv3, được viết bằng kiến trúc Darknet. Glenn Jocher đã sử dụng phiên bản YOLOv3 trong PyTorch với một số thay đổi nhỏ và đặt tên nó là YOLOv5. Kiến trúc YOLOv5 sau đó đã được điều chỉnh để phát triển YOLOv8.

### 1.3.5. Các tính năng chính của YOLOv8

Kiến trúc xương sống và cổ nâng cao: YOLOv8 tích hợp một kiến trúc xương sống và cổ hiện đại, điều này cải thiện đáng kể hiệu suất trích xuất tính năng và phát hiện đối tượng.

Chia tách không neo Ultralytics Đầu: Mô hình YOLOv8 áp dụng phân chia không neo Ultralytics Đầu, giúp cải thiện độ chính xác và hiệu quả trong quá trình phát hiện so với các phương pháp tiếp cận dựa trên neo.

Tối ưu hóa sự đánh đổi độ chính xác-tốc độ: YOLOv8 tập trung vào việc duy trì sự cân bằng tối ưu giữa độ chính xác và tốc độ. Điều này làm cho mô hình phù hợp cho các tác vụ phát hiện đối tượng theo thời gian thực trong nhiều lĩnh vực ứng dụng khác nhau.

Nhiều mô hình Pre-training: YOLOv8 cung cấp một loạt các mô hình được đào tạo trước để phù hợp với các nhiệm vụ và yêu cầu hiệu suất khác nhau. Điều này giúp người dùng dễ dàng tìm thấy mô hình phù hợp nhất cho trường hợp sử dụng cụ thể của họ.

### 1.3.6. YOLOv8 có ưu điểm và nhược điểm gì?

Ưu điểm:

* Hiệu suất cải thiện: YOLOv8 có thể cải thiện hiệu suất của mô hình nhận diện đối tượng, bao gồm cả tốc độ xử lý và độ chính xác, thông qua việc sử dụng các kỹ thuật tối ưu hóa mới và cải tiến trong kiến trúc mạng.
* Hỗ trợ cho đối tượng nhỏ và chồng chéo: YOLOv8 có thể cải thiện khả năng nhận diện các đối tượng nhỏ và các đối tượng chồng chéo, tăng cường độ chính xác và sự linh hoạt của mô hình trong các tình huống phức tạp.
* Tính linh hoạt và tuỳ chỉnh: YOLOv8 có thể cung cấp các tính năng cho phép người dùng tuỳ chỉnh mô hình và điều chỉnh các siêu tham số theo nhu cầu cụ thể của họ, bao gồm cơ chế tự động hóa để tối ưu hóa hiệu suất của mô hình trên các tập dữ liệu cụ thể.
* Hỗ trợ nhiều nền tảng và ứng dụng: YOLOv8 có thể được thiết kế để hỗ trợ một loạt các nền tảng và ứng dụng, từ máy tính cá nhân đến các thiết bị nhúng và các ứng dụng thời gian thực.
* Tích hợp các kỹ thuật mới: YOLOv8 có thể tích hợp các kỹ thuật mới trong lĩnh vực nhận diện đối tượng như attention mechanisms, self-supervised learning, hoặc các kỹ thuật học sâu tiên tiến khác để cải thiện hiệu suất của mô hình.

Nhược điểm:

* Yêu cầu tài nguyên tính toán: Cải tiến hiệu suất có thể đồng nghĩa với việc đòi hỏi nhiều tài nguyên tính toán hơn, có thể làm giảm khả năng triển khai trên các thiết bị có tài nguyên hạn chế.
* Độ chính xác phụ thuộc vào dữ liệu: YOLOv8 yêu cầu một lượng lớn dữ liệu huấn luyện đa dạng để đạt được hiệu suất tốt nhất. Thiếu dữ liệu hoặc dữ liệu không đa dạng có thể làm giảm độ chính xác của mô hình.
* Khó khăn trong việc đặt siêu tham số: Một số tính năng tự động hoá có thể gặp khó khăn trong việc đặt các siêu tham số cho YOLOv8, đòi hỏi người dùng có kiến thức chuyên sâu về mạng neural và lĩnh vực nhận diện đối tượng.

### 1.3.7. YOLOv8 so với các phiên bản khác?

Giống nhau:

* Phương pháp cơ bản: YOLOv8 có thể vẫn sử dụng cùng một phương pháp cơ bản "You Only Look Once" (YOLO) như các phiên bản YOLO trước đó, trong đó đồng thời nhận diện và phân loại các đối tượng trong một lần chạy mạng neural trên toàn bức ảnh.
* Sử dụng bounding boxes: YOLOv8 có thể vẫn sử dụng bounding boxes để định vị vị trí của các đối tượng trong hình ảnh, giống như các phiên bản YOLO trước đó.
* Mã nguồn mở: YOLOv8 có thể vẫn là mã nguồn mở, cho phép cộng đồng nghiên cứu và phát triển tiếp tục cải thiện và tùy chỉnh mô hình.

Khác nhau:

* Hiệu suất và độ chính xác: YOLOv8 có thể được cải thiện về hiệu suất và độ chính xác so với các phiên bản YOLO trước đó, thông qua việc sử dụng các kỹ thuật tối ưu hóa mới và cải tiến trong kiến trúc mạng.
* Tính linh hoạt và tuỳ chỉnh: YOLOv8 có thể cung cấp nhiều tính năng linh hoạt hơn và tuỳ chỉnh hơn cho phép người dùng điều chỉnh mô hình theo nhu cầu cụ thể của họ, có thể thông qua các cơ chế tự động hóa và điều chỉnh siêu tham số.
* Hỗ trợ cho đối tượng nhỏ và chồng chéo: YOLOv8 có thể cải thiện khả năng nhận diện các đối tượng nhỏ và các đối tượng chồng chéo, tăng cường độ chính xác và sự linh hoạt của mô hình trong các tình huống phức tạp.
* Tích hợp các kỹ thuật mới: YOLOv8 có thể tích hợp các kỹ thuật mới trong lĩnh vực nhận diện đối tượng như attention mechanisms, self-supervised learning, hoặc các kỹ thuật học sâu tiên tiến khác để cải thiện hiệu suất của mô hình.

## 1.4. Ultralytics là gì?



Ultralytics, một công ty công nghệ có trụ sở tại Los Angeles, California, chuyên về phát triển phần mềm và công nghệ trí tuệ nhân tạo (AI). Công ty tập trung vào các lĩnh vực như thị giác máy tính, học sâu và học máy.

Một trong những sản phẩm nổi bật của Ultralytics là YOLOv5, một mô hình nhận diện đối tượng dựa trên phương pháp "You Only Look Once" (YOLO). YOLOv5 đã thu hút sự chú ý lớn từ cộng đồng nhờ vào hiệu suất cao và tính dễ sử dụng, và đã được áp dụng rộng rãi trong nhiều ứng dụng thực tế.

Ngoài ra, Ultralytics còn cung cấp các công cụ và tài nguyên khác trong lĩnh vực trí tuệ nhân tạo, bao gồm mã nguồn mở, hướng dẫn và tư vấn, nhằm hỗ trợ các nhà phát triển và nghiên cứu trong việc triển khai và phát triển các ứng dụng trí tuệ nhân tạo.

Điều này làm cho Ultralytics trở thành một nguồn tài nguyên quan trọng và đáng tin cậy đối với cộng đồng AI, đồng thời thúc đẩy sự tiến bộ trong lĩnh vực này.

## 1.5. Tracking là gì?



Bài toán theo dõi đối tượng trong thời gian thực đòi hỏi khả năng quan sát và theo dõi sự di chuyển của đối tượng qua các khung hình liên tiếp trong một video. Khác với bài toán nhận diện đối tượng, trong đó chỉ cần xác định có mặt của đối tượng trong mỗi khung hình, bài toán theo dõi đối tượng cần phải xác định và theo dõi đối tượng từ khung hình này sang khung hình khác, giữ cho đối tượng được định danh theo thời gian.

Có nhiều phương pháp để giải quyết bài toán theo dõi đối tượng, từ các phương pháp cổ điển dựa trên việc áp dụng các thuật toán xử lý ảnh và computer vision, đến các phương pháp tiên tiến sử dụng học sâu và mạng neural để học và dự đoán quỹ đạo di chuyển của đối tượng.

Việc theo dõi đối tượng trong thời gian thực có ứng dụng rộng rãi trong nhiều lĩnh vực như giám sát an ninh, theo dõi giao thông, trí tuệ nhân tạo cho xe tự lái, và cả trong công nghiệp giải trí và sản xuất video.

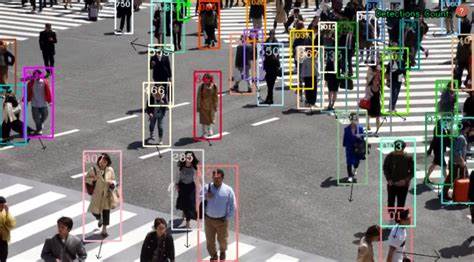
### 1.5.1. Phân loại Tracking?

Object Tracking có thể chia thành 2 cách tiếp cận chính:

Single Object Tracking (SOT): Cái tên nói lên tất cả, Single Object Tracking tập trung vào việc theo dõi một đối tượng duy nhất trong toàn bộ video. Và tất nhiên, để biết được cần theo dõi đối tượng nào, việc cung cấp một bounding box từ ban đầu là việc bắt buộc phải có.



Mutiple Object Tracking (MOT): Mutliple Object Tracking hướng tới các ứng dụng có tính mở rộng cao hơn. Bài toán cố gắng phát hiện đồng thời theo dõi tất cả các đối tượng trong tầm nhìn, kể cả các đối tượng mới xuất hiện trong video. Vì điểu này, MOT thường là những bài toán khó hơn SOT và nhận được rất nhiều sự quan tâm của giới nghiên cứu.



Ngoài việc tiếp cận vấn đề, các phương pháp giải quyết bài toán theo dõi đối tượng cũng được phân chia rất đa dạng. Các phương pháp phổ biến nhất bao gồm:

1. Theo dõi trực tuyến (Online Tracking): Phương pháp này chỉ sử dụng frame hiện tại và frame trước đó để thực hiện theo dõi. Mặc dù có thể làm giảm độ chính xác của thuật toán, nhưng nó phản ánh đúng cách vấn đề được xử lý trong thực tế, khi tính "trực tuyến" là cần thiết.

2. Theo dõi ngoại tuyến (Offline Tracking): Các phương pháp ngoại tuyến thường sử dụng toàn bộ frame của video, do đó thường đạt được độ chính xác cao hơn nhiều so với theo dõi trực tuyến.

Bên cạnh đó, còn có phân loại dựa trên:

- Theo dõi dựa trên phát hiện (Detection-based Tracking): Tập trung vào việc liên kết chặt chẽ giữa phát hiện đối tượng và theo dõi đối tượng, từ đó dựa vào kết quả của việc phát hiện để theo dõi đối tượng qua các frame.

- Theo dõi không cần phát hiện (Detection Free Tracking): Xem video như một dạng dữ liệu chuỗi, từ đó, áp dụng những phương pháp dành riêng cho "chuỗi" như RNN (Mạng nơ-ron hồi quy), LSTM (Mạng nơ-ron dài-ngắn hạn), và các phương pháp khác.

## 1.6. Ngôn ngữ lập trình, thư viện và công cụ hỗ trợ

### 1.6.1. Python



Python là một ngôn ngữ lập trình cao cấp, có cú pháp dễ đọc và dễ hiểu. Nó được phát triển vào những năm 1990 và đã trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới. Python được thiết kế để tối ưu hóa sự đọc và viết mã, giúp người lập trình dễ dàng tạo ra các ứng dụng và phần mềm.

### 1.6.2. Ưu nhược điểm của Python



Ưu điểm:

1. Dễ đọc và dễ học

2. Giảm chi phí bảo trì

3. Tránh tác hại từ lỗi phần mềm

4. Khả năng ứng dụng rộng rãi

5. Quản lý bộ nhớ

6. Đơn giản và nhanh chóng

7. Mã hóa không đồng bộ

8. Tích hợp với các ngôn ngữ khác

9. Tích hợp ứng dụng doanh nghiệp

Nhược điểm:

1. Tốc độ thực thi chậm

2. Tiêu thụ bộ nhớ lớn

3. Không thích hợp cho việc phát triển trò chơi và ứng dụng di động

4. Hạn chế của nhà phát triển

5. Phát hiện lỗi trong mã

6. Quyền truy cập cơ sở dữ liệu

7. Hạn chế thiết kế

8. Khó kiểm tra

### 1.6.3. Thư viện OpenCV



OpenCV (Open Source Computer Vision Library) không chỉ là một thư viện mã nguồn mở phổ biến mà còn là một công cụ mạnh mẽ được sử dụng rộng rãi trong lĩnh vực thị giác máy tính và xử lý hình ảnh. Với sự tích hợp linh hoạt và các thuật toán tiên tiến, OpenCV cung cấp cho các nhà phát triển và nghiên cứu một nền tảng mạnh mẽ để thực hiện các dự án và nghiên cứu trong các lĩnh vực như:

1. Nhận diện khuôn mặt: OpenCV cung cấp các công cụ để nhận diện, phân loại và theo dõi khuôn mặt trong các hình ảnh và video, giúp trong việc xây dựng các hệ thống nhận diện khuôn mặt hoặc ứng dụng về nhận diện người dùng.

2. Phát hiện đối tượng: Thư viện cung cấp các thuật toán phát hiện đối tượng hiệu quả như phát hiện xe, người, đồ vật và nhiều hơn nữa, hỗ trợ trong việc xây dựng các hệ thống giám sát, xe tự lái và các ứng dụng khác.

3. Xử lý hình ảnh và video: OpenCV cung cấp các công cụ và thuật toán để xử lý hình ảnh và video, bao gồm lọc ảnh, biến đổi hình ảnh, trích xuất đặc trưng, đối sánh hình ảnh và nhiều thao tác khác.

4. Thị giác máy tính: Thư viện hỗ trợ các nghiên cứu và ứng dụng trong lĩnh vực thị giác máy tính, bao gồm việc xây dựng hệ thống phân loại hình ảnh, phát hiện vật thể và nhận dạng hình ảnh.

Với sự linh hoạt, hiệu suất và tính ổn định, OpenCV đã trở thành một công cụ quan trọng không chỉ cho các nhà phát triển phần mềm mà còn cho các nhà nghiên cứu và chuyên gia trong lĩnh vực thị giác máy tính và xử lý hình ảnh.

### 1.6.4. Ưu nhược điểm của OpenCV

Ưu điểm:

* Miễn phí và mã nguồn mở
* Hỗ trợ đa nền tảng
* Bộ công cụ mạnh mẽ
* Tích hợp dễ dàng
* Cộng đồng lớn và tài liệu phong phú

Nhược điểm:

* Cộng đồng lớn và tài liệu phong phú
* Khả năng tích hợp với các ngôn ngữ khác nhau
* Hiệu năng
* Hạn chế trong việc xử lý ảnh nâng cao

### 1.6.5. Những khó khăn trong việc nhận dạng đối tượng?

Biến đổi đối tượng: Đối tượng có thể xuất hiện ở nhiều hình dạng, kích thước, màu sắc và góc độ khác nhau. Điều này làm cho việc nhận dạng chính xác trở nên khó khăn, đặc biệt là khi môi trường chiếu sáng thay đổi hoặc có nhiều nhiễu.

Đối tượng che khuất: Đối tượng có thể bị che khuất bởi các vật thể khác trong cảnh quan, gây ra sự khó khăn trong việc nhận dạng. Điều này đặc biệt quan trọng trong các ứng dụng như nhận dạng giao thông hoặc nhận diện người trong các tình huống phức tạp.

Đối tượng mờ hoặc không rõ ràng: Đối tượng có thể bị mờ hoặc không rõ ràng trong các hình ảnh có độ phân giải thấp hoặc bị nhiễu. Điều này làm giảm khả năng nhận dạng chính xác của các hệ thống.

Đa dạng về đối tượng: Các hệ thống nhận dạng đối tượng cần phải đối mặt với đa dạng về loại đối tượng, từ người, xe đến động vật và vật thể không gian. Điều này đặt ra thách thức về việc phát triển các mô hình nhận dạng có khả năng phân loại rộng rãi.

Thiếu dữ liệu huấn luyện: Việc thu thập dữ liệu đủ lớn và đa dạng để huấn luyện một mô hình nhận dạng đối tượng có thể là một thách thức. Thiếu dữ liệu đủ lớn và đa dạng có thể dẫn đến hiệu suất kém của mô hình.

Tài nguyên tính toán: Các mô hình nhận dạng đối tượng thường đòi hỏi nhiều tài nguyên tính toán để triển khai, đặc biệt là trong các ứng dụng thời gian thực hoặc trên các thiết bị có tài nguyên hạn chế.

Bảo mật và quyền riêng tư: Việc nhận dạng đối tượng có thể đặt ra các vấn đề về quyền riêng tư và bảo mật, đặc biệt khi áp dụng trong các hệ thống giám sát và nhận dạng người.

# CHƯƠNG 2

# Phân tích yêu cầu

## 2.1. Yêu cầu đề tài

Đề tài về nhận diện người trong trung tâm thương mại là một lĩnh vực quan trọng trong lĩnh vực thị giác máy tính và trí tuệ nhân tạo. Dưới đây là một phân tích cụ thể về yêu cầu và thách thức của đề tài này:

Nhận diện người: Nhiệm vụ chính là phát hiện và nhận diện người trong hình ảnh hoặc video, bao gồm cả việc xác định vị trí và đặc điểm của họ.

Phân loại hành động hoặc hành vi: Ngoài việc nhận diện người, có thể yêu cầu phân loại hành động hoặc hành vi của họ, chẳng hạn như đi, đứng, hoặc đeo khẩu trang.

Theo dõi người qua thời gian: Một yêu cầu khác có thể là theo dõi các người qua các khung hình liên tiếp, từ đó xác định hành vi di chuyển hoặc quan hệ giữa các người trong trung tâm thương mại.

## 2.2. Phân tích yêu cầu bài toán

Thu thập và chuẩn bị dữ liệu: Thu thập một tập dữ liệu đủ lớn và đa dạng về các hình ảnh hoặc video chứa các đối tượng mà bạn muốn nhận diện. Dữ liệu cần được gắn nhãn chính xác với các vị trí và lớp tương ứng của các đối tượng.

Xây dựng mô hình nhận diện đối tượng: Sử dụng một trong các phương pháp nhận diện đối tượng như YOLO (You Only Look Once), SSD (Single Shot Multibox Detector), Faster R-CNN (Faster Region-based Convolutional Neural Network), hoặc các kiến trúc mạng neural khác để xây dựng mô hình nhận diện đối tượng.

Huấn luyện mô hình: Sử dụng dữ liệu đã được gắn nhãn, huấn luyện mô hình nhận diện đối tượng trên máy tính hoặc cụm máy tính phân tán để tối ưu hóa các tham số của mô hình.

Đánh giá và fine-tuning: Đánh giá hiệu suất của mô hình trên tập dữ liệu kiểm tra và điều chỉnh các siêu tham số để cải thiện độ chính xác và độ tin cậy của mô hình.

Triển khai và tích hợp: Triển khai mô hình đã huấn luyện vào các ứng dụng thực tế, bao gồm việc tích hợp vào các hệ thống nhúng, ứng dụng di động, hoặc hệ thống giám sát.

Kiểm tra và cải thiện liên tục: Tiếp tục kiểm tra và cải thiện mô hình dựa trên phản hồi từ việc triển khai và sử dụng thực tế, cũng như việc thu thập thêm dữ liệu hoặc thay đổi cấu trúc mô hình khi cần thiết.

## 2.3. Thách thức trong đề tài

Đa dạng về điều kiện ánh sáng và nền nhiễu: Trung tâm thương mại thường có nhiều điều kiện ánh sáng khác nhau và có nhiều nguồn nhiễu, như ánh sáng mạnh từ cửa sổ hoặc ánh sáng yếu trong các khu vực tối.

Biến đổi hình dạng và góc nhìn: Người trong trung tâm thương mại có thể mặc quần áo khác nhau, đeo khẩu trang hoặc đổi góc nhìn, làm cho việc nhận diện trở nên khó khăn.

Mật độ đông đúc: Trong một môi trường thương mại, có thể có nhiều người đi lại cùng một lúc, tạo ra các tình huống chồng chéo và che khuất, làm giảm độ chính xác của việc nhận diện.

Quản lý dữ liệu: Dữ liệu hình ảnh hoặc video từ trung tâm thương mại có thể lớn và phức tạp, đòi hỏi các phương pháp hiệu quả để xử lý và phân tích.

Quản lý quyền riêng tư: Việc nhận diện người trong một môi trường công cộng như trung tâm thương mại đặt ra các vấn đề về quyền riêng tư và bảo mật, yêu cầu các biện pháp đảm bảo tính riêng tư và tuân thủ các quy định pháp luật.

# CHƯƠNG 3

# Xây dựng mô hình

## 3.1. Quy trình thực nghiệm mô hình

Bước 1: Import các thư viện cần thiết:

Bước 2: Khởi tạo mô hình YOLO: Sử dụng file trọng số 'yolov8s.pt' để tạo mô hình.

Bước 3: Mở video hoặc stream video từ một nguồn nhất định, như một file video hoặc một luồng trực tuyến từ YouTube.

Bước 4: Định nghĩa hàm callback RGB để lấy giá trị màu RGB khi di chuột qua video.

Bước 5: Đọc danh sách các lớp từ file "coco.txt".

Bước 6: Khởi tạo một đối tượng Tracker cho việc theo dõi các đối tượng trong video.

Bước 7: Xác định vùng quan tâm trong video bằng cách xác định các điểm (đỉnh của một hình đa giác) area và area2 mà các đối tượng sẽ được theo dõi khi di chuyển qua đó.

Bước 8: Trong vòng lặp vô hạn, tiến hành các bước sau:

1. Đọc frame từ video.
2. Đưa frame qua mô hình YOLO để phát hiện các đối tượng.
3. Xác định vị trí của các đối tượng (vị trí và lớp) và lưu trữ chúng trong danh sách list.
4. Cập nhật các đối tượng đã được theo dõi trong video trước đó bằng cách sử dụng module theo dõi.
5. Vẽ hình chữ nhật và hiển thị lớp của mỗi đối tượng trên frame.
6. Theo dõi việc di chuyển của các đối tượng qua các vùng quan tâm được xác định trước đó và đếm số lượng đối tượng đi qua từng vùng.
7. Vẽ các đường và thông tin thống kê trên frame.
8. Hiển thị frame lên cửa sổ 'RGB'.
9. Nếu phím ESC được nhấn thì thoát khỏi vòng lặp.

Bước 9: Giải phóng tài nguyên và đóng tất cả cửa sổ khi kết thúc.

## 3.2. Các thư viện được sử dụng

Import cv2: Thư viện OpenCV cho xử lý hình ảnh và video.

Import pandas: Thư viện cho phân tích dữ liệu.

Import numpy: Thư viện cho tính toán khoa học.

ultralytics.YOLO: Một phiên bản của mô hình YOLO.

Import cvzone: Một thư viện hỗ trợ OpenCV cho việc vẽ văn bản và hình ảnh trên video.

vidgear.gears.CamGear: Thư viện cho việc lấy video từ webcam hoặc URL.

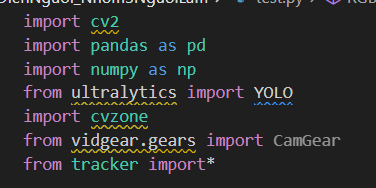
tracker: Một module tự định nghĩa (không được hiển thị ở đây) cho việc theo dõi các đối tượng.

# 

# CHƯƠNG 4

# Thực nghiệm mô hình

## 4.1. Cài đặt các phụ thuộc



import cv2: Import thư viện OpenCV để làm việc với hình ảnh và video.

import pandas as pd: Import thư viện pandas để thao tác và phân tích dữ liệu dưới dạng DataFrame.

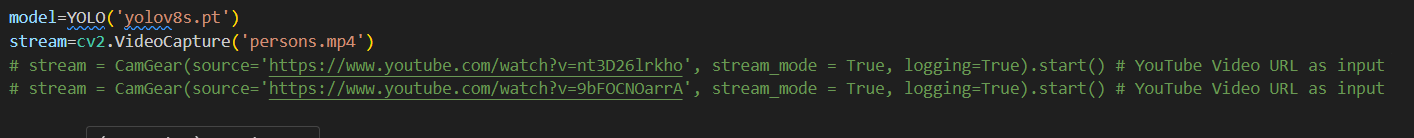
numpy: Thư viện cho các phép toán số học.

from ultralytics import YOLO: Import class YOLO từ thư viện Ultralytics để sử dụng mô hình YOLO.

import cvzone: Thư viện thị giác máy tính cấp cao dựa trên OpenCV.

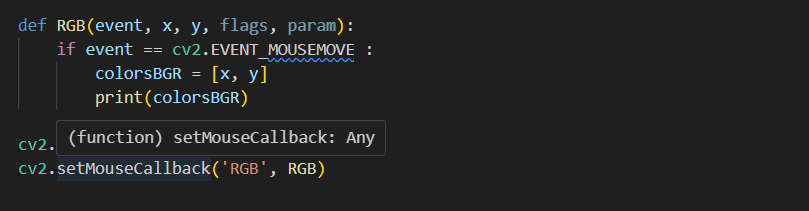
from vidgear.gears import CamGear: Mô-đun từ VidGear để truy cập các luồng video.

from tracker import\*: Mô-đun theo dõi đối tượng do bạn tự triển khai.



model: Khởi tạo mô hình YOLO với trọng số từ tệp yolov8s.pt.

stream: Đọc video từ tệp persons.mp4. Hai dòng sau là ví dụ về cách sử dụng CamGear để lấy luồng video từ YouTube.



RGB: Hàm này được gọi khi có sự kiện chuột. Nếu chuột di chuyển, nó in ra tọa độ của con trỏ chuột.

cv2.namedWindow: Tạo cửa sổ có tên 'RGB'.

cv2.setMouseCallback: Gán hàm 'RGB' cho cửa sổ 'RGB' để xử lý sự kiện chuột.



classnames: Danh sách tên các lớp đối tượng.

counter\_down: Danh sách đếm số đối tượng đi xuống.

counter\_up: Danh sách đếm số đối tượng đi lên.

down và up: Từ điển lưu tọa độ y của các đối tượng khi chúng đi qua vùng xác định.

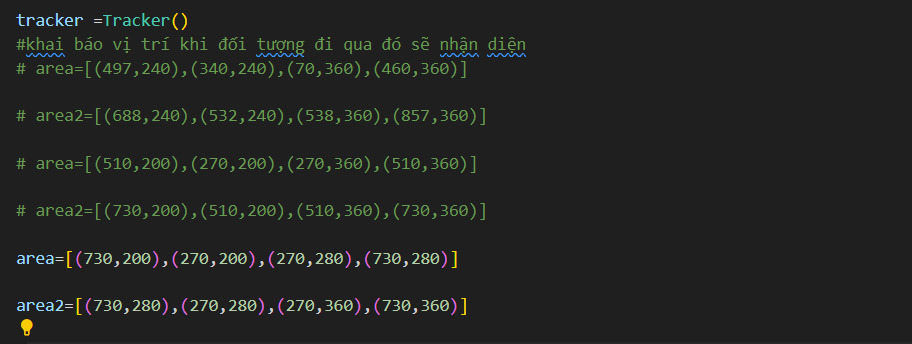


my\_file: Mở tệp coco.txt chứa tên các lớp đối tượng.

data: Đọc nội dung của tệp.

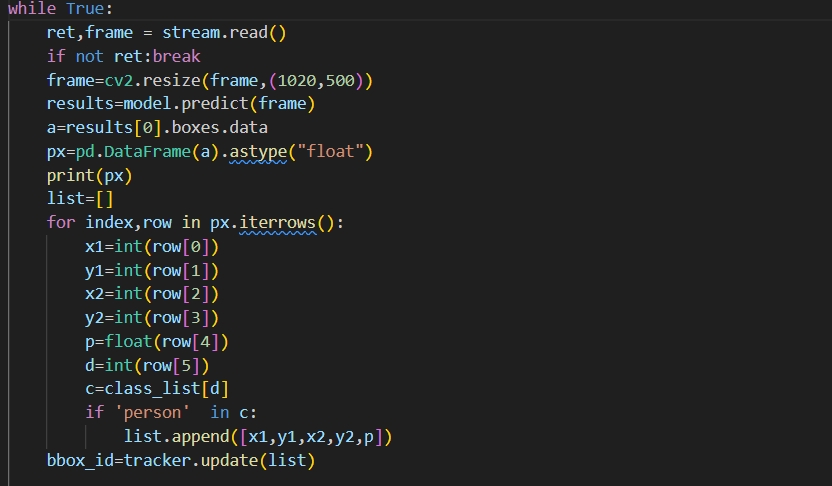
class\_list: Chia nội dung tệp thành danh sách tên các lớp.

count, count\_up, count\_down: Các biến đếm.



tracker: Khởi tạo đối tượng Tracker.

area và area2: Khai báo vùng để nhận diện khi đối tượng đi qua.



while True: Vòng lặp để xử lý từng khung hình video.

ret, frame: Đọc một khung hình từ video.

if not ret: break: Nếu không đọc được khung hình, thoát khỏi vòng lặp.

frame = cv2.resize: Thay đổi kích thước khung hình.

results = model.predict(frame): Dự đoán các đối tượng trong khung hình sử dụng mô hình YOLO.

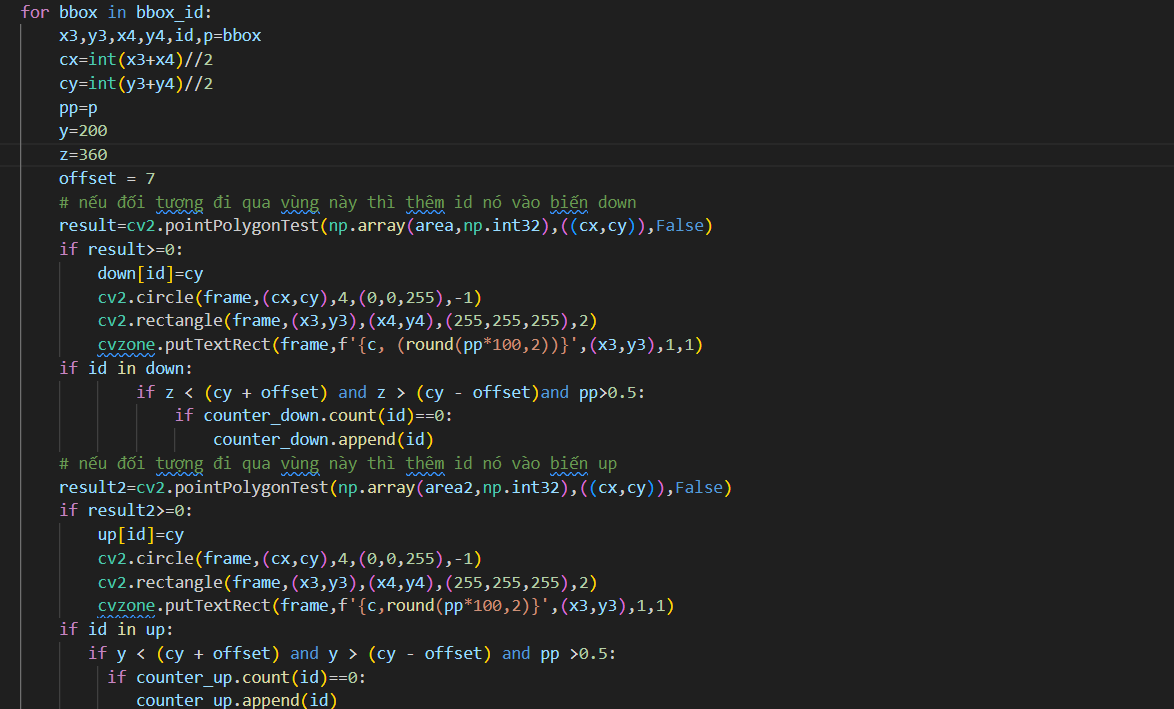
a = results[0].boxes.data: Lấy dữ liệu hộp giới hạn từ kết quả dự đoán.

px = pd.DataFrame(a).astype("float"): Chuyển dữ liệu thành DataFrame của pandas và đổi kiểu dữ liệu thành float.

for index, row in px.iterrows(): Duyệt qua từng hàng trong DataFrame.

list.append([x1, y1, x2, y2, p]): Thêm thông tin hộp giới hạn vào danh sách nếu đối tượng là 'person'.

bbox\_id = tracker.update(list): Cập nhật danh sách hộp giới hạn vào đối tượng tracker.



for bbox in bbox\_id: Duyệt qua từng hộp giới hạn và thông tin ID từ tracker.

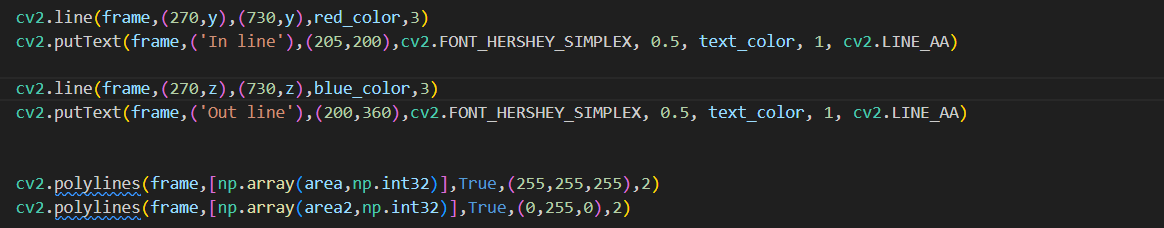
cx, cy: Tọa độ trung tâm của hộp giới hạn.

result = cv2.pointPolygonTest(np.array(area, np.int32), (cx, cy), False): Kiểm tra nếu trung tâm đối tượng nằm trong vùng area.

if result >= 0: Nếu đúng, thêm ID của đối tượng vào down và vẽ vòng tròn, hộp giới hạn trên khung hình.

if id in down: Kiểm tra nếu ID đối tượng trong down, tăng biến đếm counter\_down khi đối tượng đi qua vùng down.

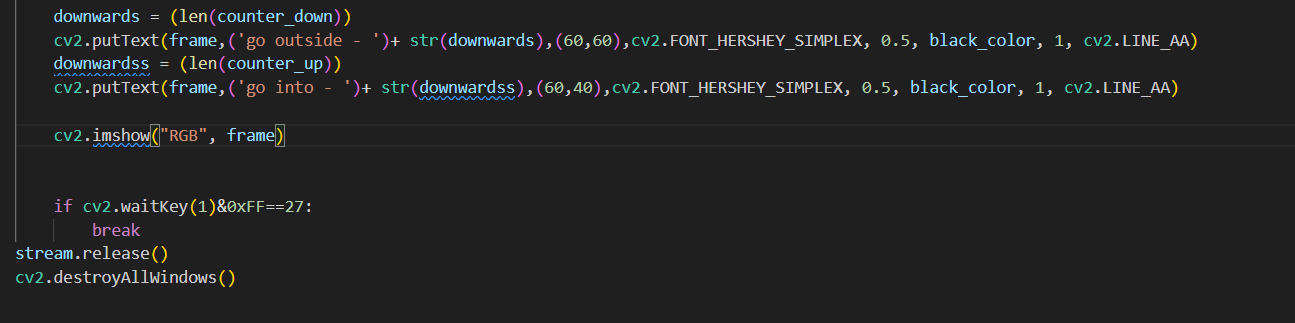
result2 = cv2.pointPolygonTest(np.array(area2, np.int32), (cx, cy), False): Tương tự kiểm tra cho vùng area2 và tăng biến đếm counter\_up.



cv2.line, cv2.putText: Vẽ hai đường thẳng và thêm văn bản

cv2.polylines(frame, [np.array(area, np.int32)], True, (255, 255, 255), 2): Vẽ đa giác với tọa độ được xác định trong area

cv2.polylines(frame, [np.array(area2, np.int32)], True, (0, 255, 0), 2): Vẽ đa giác màu xanh lá cây với tọa độ được xác định trong area2



downwards = len(counter\_down): Đếm số đối tượng đi ra ngoài.

downwardss = len(counter\_up): Đếm số đối tượng đi vào trong

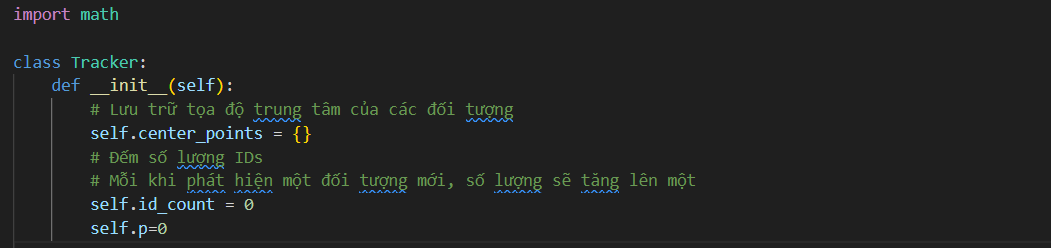
cv2.imshow("RGB", frame): Hiển thị khung hình hiện tại.

if cv2.waitKey(1) & 0xFF == 27: Nếu nhấn phím 'Esc', thoát khỏi vòng lặp.

stream.release(): Giải phóng video stream.

cv2.destroyAllWindows(): Đóng tất cả các cửa sổ OpenCV.

## 4.2. Tracker

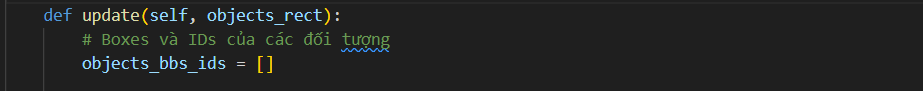


Thêm thư viện math để sử dụng hàm toán học.

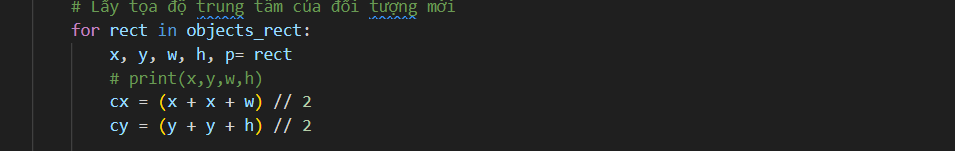
self.center\_points: Từ điển lưu trữ tọa độ trung tâm của các đối tượng.

self.id\_count: Biến đếm để tạo ID duy nhất cho mỗi đối tượng mới.

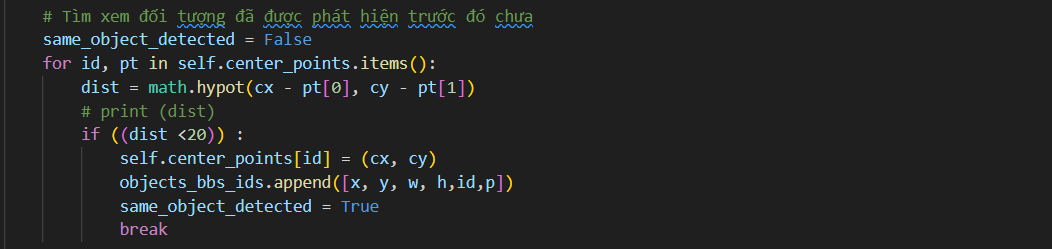
self.p: Biến lưu trữ xác suất (không được sử dụng nhiều trong mã nguồn hiện tại).



objects\_rect: Danh sách các hình chữ nhật chứa tọa độ của các đối tượng.

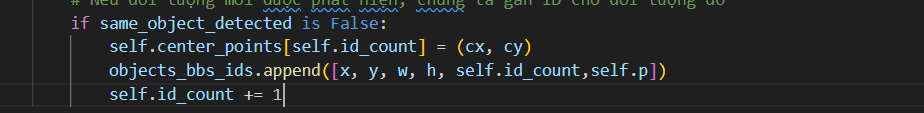


Tọa độ trung tâm cx và cy được tính từ các tọa độ của hình chữ nhật.



Tính khoảng cách từ tọa độ trung tâm hiện tại đến tọa độ trung tâm của các đối tượng đã được phát hiện trước đó.

Nếu khoảng cách nhỏ hơn 20, coi như đối tượng đã được phát hiện và cập nhật tọa độ trung tâm.

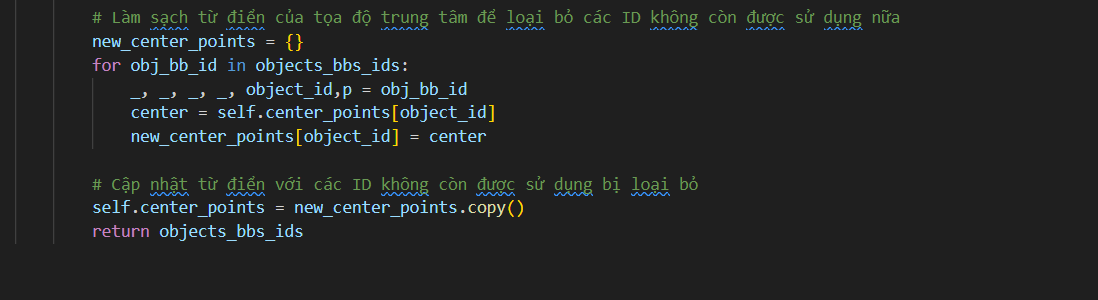


Nếu same\_object\_detected là False, điều đó có nghĩa là đối tượng này chưa được phát hiện trước đó và cần gán một ID mới.

self.center\_points là một từ điển lưu trữ các ID của đối tượng và tọa độ trung tâm tương ứng của chúng.

objects\_bbs\_ids.append([x, y, w, h, self.id\_count, self.p]): Thêm thông tin của đối tượng mới (tọa độ góc trái trên x, y, chiều rộng w, chiều cao h, ID của đối tượng và xác suất self.p) vào danh sách objects\_bbs\_ids.

self.id\_count += 1: Tăng giá trị ID lên 1 để chuẩn bị cho đối tượng mới tiếp theo.



new\_center\_points = {}: Khởi tạo một từ điển rỗng để lưu trữ các điểm trung tâm mới.

Lặp qua từng phần tử trong objects\_bbs\_ids,Trích xuất object\_id và p từ mỗi tuple. Có vẻ như object\_id được sử dụng làm khóa để truy cập vào các điểm trung tâm từ self.center\_points.

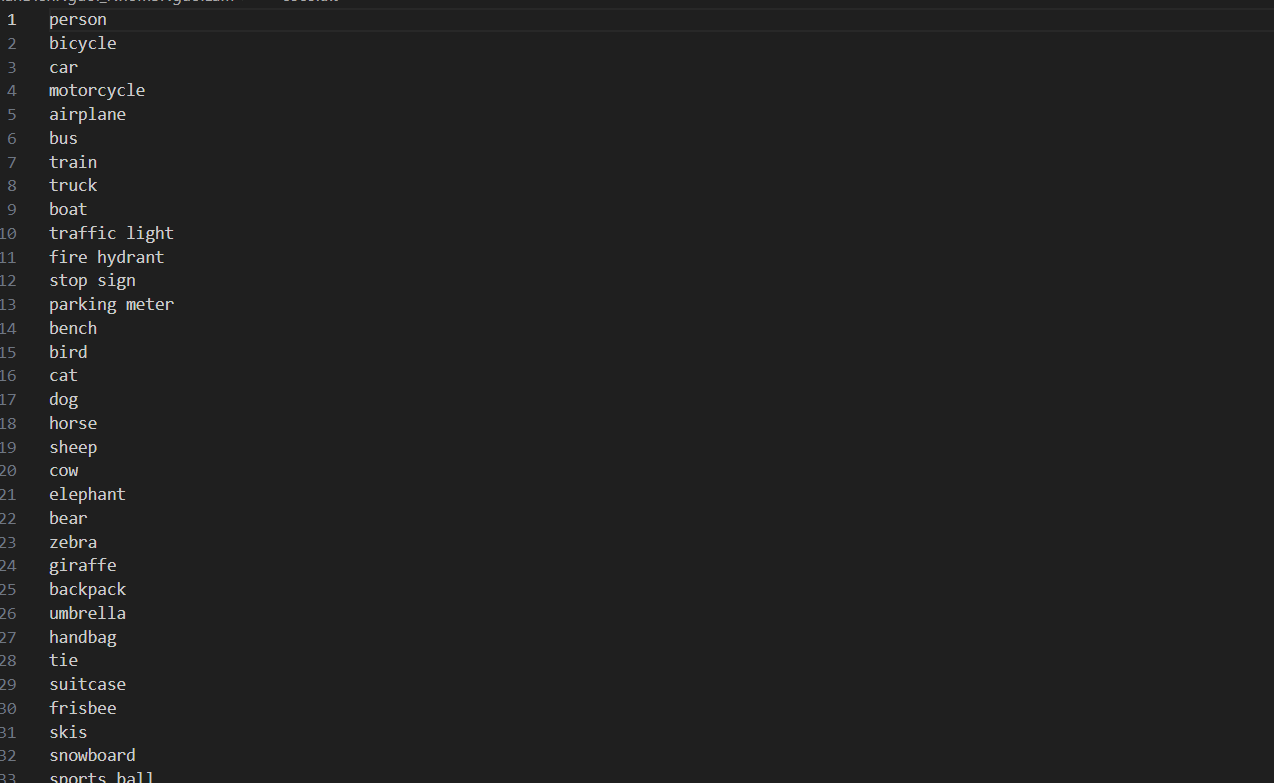
Gán từ điển đã sao chép new\_center\_points vào self.center\_points, hiệu quả là cập nhật self.center\_points.

Cuối cùng, nó trả về objects\_bbs\_ids.

## 4.3. Video để nhận diện



## 4.4. Đối tượng để huấn luyện

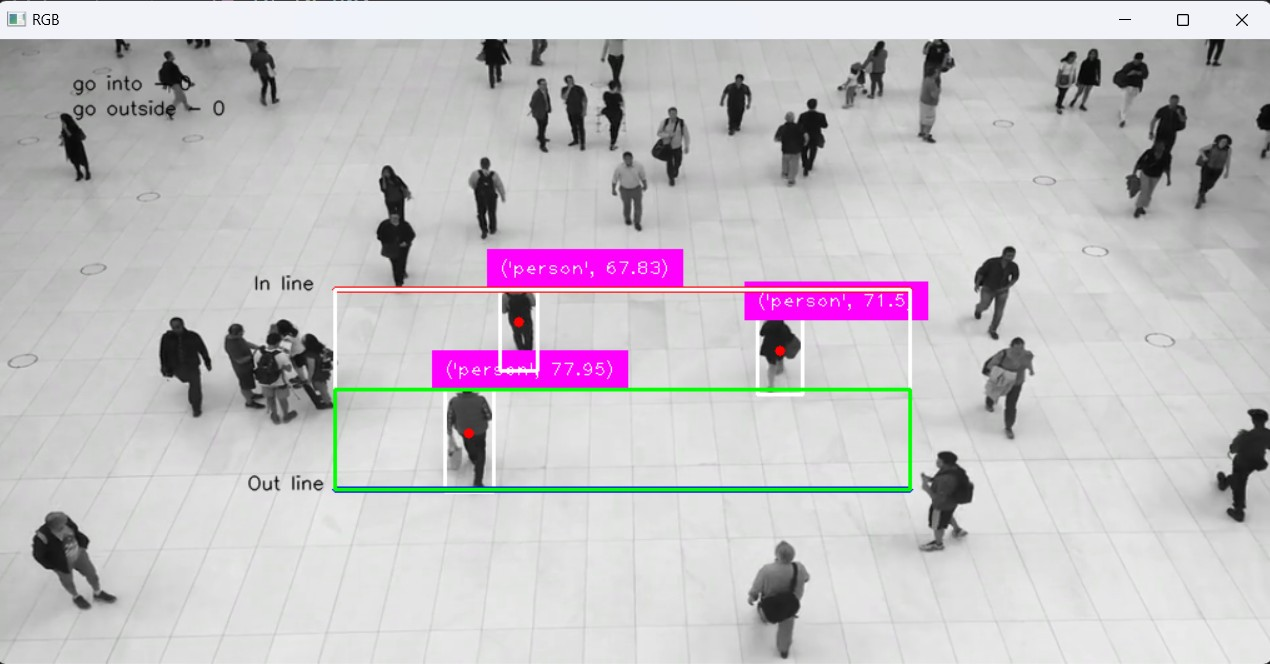


# 

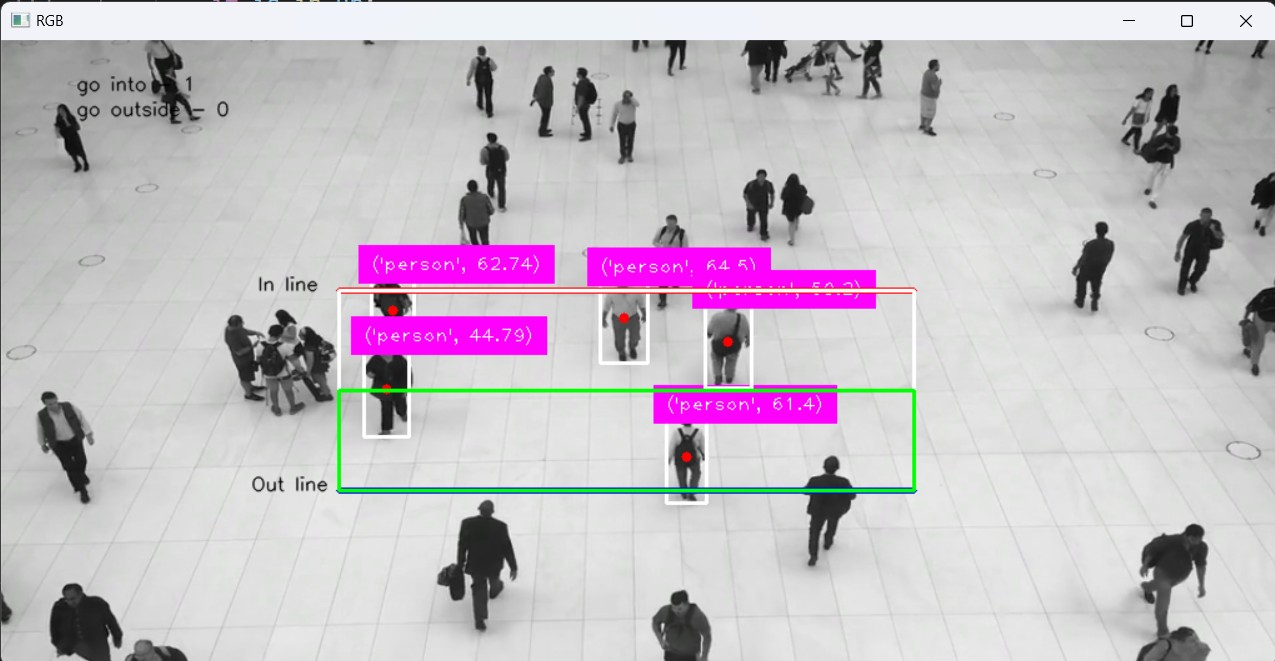
# CHƯƠNG 5

# Kết luận và hướng phát triển

## 5.1. Kết quả đạt được









## 5.2. Hạn chế của đề tài

Hiệu suất: YOLO có thể đối mặt với vấn đề hiệu suất khi triển khai trong môi trường thương mại, đặc biệt là nếu cần xử lý video trực tiếp hoặc nhiều luồng video cùng một lúc. Việc đảm bảo hiệu suất đủ cao để xử lý thời gian thực có thể là một thách thức.

Dữ liệu huấn luyện: Để đạt được hiệu suất tốt nhất, bạn cần một tập dữ liệu huấn luyện đủ lớn và đa dạng về người và điều kiện môi trường. Thu thập và chuẩn bị dữ liệu có thể là một quá trình tốn kém và tốn thời gian.

Độ chính xác: Mặc dù YOLO có khả năng nhận diện đối tượng nhanh chóng, nhưng độ chính xác của nó có thể không cao bằng các mô hình phức tạp hơn, đặc biệt là trong các tình huống phức tạp như ánh sáng yếu, che khuất hoặc đối tượng nhỏ.

Bảo mật và quyền riêng tư: Việc sử dụng công nghệ nhận diện người trong môi trường thương mại đặt ra các vấn đề liên quan đến quyền riêng tư và bảo mật dữ liệu cá nhân của khách hàng và nhân viên.

## 5.3. Hướng phát triển

Tăng cường hiệu suất: Tối ưu hóa mã và triển khai YOLO trên phần cứng mạnh mẽ hoặc sử dụng các biến thể như YOLOv4 hoặc YOLOv5 để cải thiện hiệu suất nhận diện và giảm thiểu thời gian xử lý.

Tăng cường độ chính xác: Tiếp tục tinh chỉnh mô hình YOLO trên tập dữ liệu đa dạng để cải thiện độ chính xác của việc nhận diện, đặc biệt là trong các điều kiện ánh sáng khác nhau và các tình huống phức tạp.

Phát triển giao diện người dùng: Xây dựng giao diện người dùng thân thiện để quản lý và trực quan hóa dữ liệu nhận diện người, cho phép người dùng dễ dàng tương tác và trích xuất thông tin.

Mở rộng ứng dụng: Khám phá cách sử dụng dự án nhận diện người trong các ngữ cảnh khác nhau, chẳng hạn như trong lĩnh vực bảo mật, giám sát giao thông hoặc phát hiện sự cố trong môi trường làm việc.

# TÀI LIỆU THAM KHẢO

1. Deep Learning của Ian Goodfellow, Yoshua Bengio và Aaron Courville

2. Howse, J., Joshi, P., Shrimali, V. R., & Udemy (2019), Learning OpenCV 4 Computer Vision with Python, Udemy (Khóa học trực tuyến)

3. Practical Deep Learning for Coders (fast.ai) (Khóa học trực tuyến)

4. Object Detection and Recognition in Digital Images: Theory and Practice, Bogusław Cyganek (2013).

# 