**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**KHOA: CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN**

**HỌC PHẦN: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH**

**ĐỀ TÀI SỐ 12:** **XÂY DỰNG HỆ THỐNG THEO DÕI**

**ĐỐI TƯỢNG TRONG VIDEO**

**GIẢNG VIÊN HƯỚNG DẪN: LƯƠNG THỊ HỒNG LAN**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Mã sinh viên** | **Sinh viên thực hiện** | **Lớp hành chính** |
| **1** | **20210533** | **Nguyễn Minh Thuận** | **DCCNTT12.10.2** |
| **2** | **20210487** | **Nguyễn Thị Ánh** | **DCCNTT12.10.2** |
| **3** | **20210529** | **Nguyễn Đắc Huy** | **DCCNTT12.10.2** |
| **4** | **20210521** | **Bùi Thị Trang Ánh** | **DCCNTT12.10.2** |

**Bắc Ninh, năm 2024**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**KHOA: CÔNG NGHỆ THÔNG TIN**

**BÀI TẬP LỚN**

**NHÓM: 3**

**HỌC PHẦN: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH**

**ĐỀ TÀI SỐ 12:** **XÂY DỰNG HỆ THỐNG THEO DÕI**

**ĐỐI TƯỢNG TRONG VIDEO**

**GIẢNG VIÊN HƯỚNG DẪN: LƯƠNG THỊ HỒNG LAN**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Mã sinh viên** | **Sinh viên thực hiện** | **Lớp hành chính** |
| **1** | **20210533** | **Nguyễn Minh Thuận** | **DCCNTT12.10.2** |
| **2** | **20210487** | **Nguyễn Thị Ánh** | **DCCNTT12.10.2** |
| **3** | **20210529** | **Nguyễn Đắc Huy** | **DCCNTT12.10.2** |
| **4** | **20210521** | **Bùi Thị Trang Ánh** | **DCCNTT12.10.2** |

**Bắc Ninh, năm 2024**

**Bắc Ninh, năm 2024**

|  |  |
| --- | --- |
| **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**  **KHOA CÔNG NGHỆ THÔNG TIN** | **KỲ THI KẾT THÚC HỌC PHẦN**  **HỌC KỲ 1, NĂM HỌC 2024 – 2025** |

|  |  |
| --- | --- |
| **PHIẾU CHẤM THI BÀI TẬP LỚN KẾT THÚC HỌC PHẦN**  **Mã đề thi 12: Xây dựng hệ thống theo dõi đối tượng trong video**  **Tên học phần:**  **Xử lý ảnh và thị giác máy tính**  **Lớp tín chỉ: XATGMT.03.K12.02.LH.C04.1\_LT** | |
| **Cán bộ chấm thi 1**  *(Ký và ghi rõ họ tên)*  Lương Thị Hồng Lan | **Cán bộ chấm thi 2**  *(Ký và ghi rõ họ tên)* |

| **TT** | **TIÊU CHÍ** | **THANG ĐIỂM** | **Nguyễn Minh Thuận** | **Nguyễn Thị Ánh** | **Nguyễn Đắc Huy** | **Bùi Thị Trang Ánh** |
| --- | --- | --- | --- | --- | --- | --- |
| **20210533** | **20210487** | **20210529** | **20210521** |
| **1** | **Nội dung báo cáo trên Word đầy đủ** | **3.5** |  |  |  |  |
| 1.1 | Có bố cục rõ ràng (mục lục, phần mở đầu, nội dung chính, kết luận). | 0,5 |  |  |  |  |
| 1.2 | Nội dung phân tích rõ ràng, logic. | 0,5 |  |  |  |  |
| 1.3 | Có dẫn chứng, số liệu minh họa đầy đủ. | 0,5 |  |  |  |  |
| 1.4 | Ngôn ngữ và trình bày chuẩn, không lỗi chính tả. | 0,5 |  |  |  |  |
| 1.5 | Có trích dẫn tài liệu tham khảo đúng quy cách. | 0,5 |  |  |  |  |
| 1.6 | Được trình bày chuyên nghiệp (canh lề, font chữ, khoảng cách dòng hợp lý). | 0,5 |  |  |  |  |
| 1.7 | Tài liệu đầy đủ, bám sát yêu cầu của đề bài. | 0,5 |  |  |  |  |
| **2** | **Nội dung thuyết trình đầy đủ** | **1.0** |  |  |  |  |
| 2.1 | Trình bày tự tin, phát âm rõ ràng, mạch lạc. | 0,5 |  |  |  |  |
| 2.2 | Nội dung thuyết trình đúng trọng tâm, không lan man. | 0,5 |  |  |  |  |
| **3** | **Slides báo cáo đầy đủ nội dung + Hỏi đáp** | **3.0** |  |  |  |  |
| 3.1 | Slides có bố cục rõ ràng (mở đầu, nội dung, kết luận). | 0,5 |  |  |  |  |
| 3.2 | Thiết kế slides đẹp, chuyên nghiệp (màu sắc, hình ảnh minh họa). | 0,5 |  |  |  |  |
| 3.3 | Nội dung trên slides ngắn gọn, dễ hiểu, súc tích. | 0,5 |  |  |  |  |
| 3.4 | Nội dung slides phù hợp với nội dung báo cáo. | 0,5 |  |  |  |  |
| 3.5 | Trả lời câu hỏi đầy đủ, chính xác. | 0,5 |  |  |  |  |
| 3.6 | Trả lời câu hỏi tự tin, thuyết phục. | 0,5 |  |  |  |  |
| **4** | **Code đầy đủ** | **2.5** |  |  |  |  |
| 1.1 | Code được trình bày rõ ràng, có chú thích đầy đủ. | 0,5 |  |  |  |  |
| 1.2 | Code chạy đúng, không lỗi. | 0,5 |  |  |  |  |
| 1.3 | Code tối ưu, không dư thừa. | 0,5 |  |  |  |  |
| 1.4 | Đáp ứng đầy đủ các yêu cầu chức năng theo đề bài. | 0,5 |  |  |  |  |
| 1.5 | Có tính sáng tạo hoặc cải thiện so với yêu cầu. | 0,5 |  |  |  |  |
| **TỔNG ĐIỂM BẰNG SỐ:** | | **10** |  |  |  |  |
| **TỔNG ĐIỂM BẰNG CHỮ:** | | *Mười tròn* |  |  |  |  |

MỤC LỤC

[DANH MỤC HÌNH ẢNH 7](#_Toc184583207)

[LỜI NÓI ĐẦU 8](#_Toc184583208)

[LỜI CẢM ƠN 9](#_Toc184583209)

[LỜI CAM ĐOAN 10](#_Toc184583210)

[Chương 1: Cơ sở lí thuyết 12](#_Toc184583211)

[1.1. Nhận dạng và theo dõi đối tượng là gì? 12](#_Toc184583212)

[1.2. Ứng dụng của hệ thống theo dõi đối tượng 12](#_Toc184583213)

[1.2.1 Giám sát an ninh 12](#_Toc184583214)

[1.2.2 Hệ thống giao thông thông minh 12](#_Toc184583215)

[1.2.3 Quản lý đám đông 13](#_Toc184583216)

[1.2.4 Thể thao và giải trí 13](#_Toc184583217)

[1.2.5 Quay phim và sản xuất truyền hình 13](#_Toc184583218)

[1.3. Các kỹ thuật học máy sử dụng trong hệ thống 13](#_Toc184583219)

[1.3.1. Mạng nơ-ron tích chập (Convolutional Neural Networks - CNN) 13](#_Toc184583220)

[1.3.2. YOLO (You Only Look Once) 15](#_Toc184583221)

[1.3.3. SORT và DeepSORT 16](#_Toc184583222)

[1.4. Ngôn ngữ lập trình và các thư viện 17](#_Toc184583223)

[1.4.1. Ngôn ngữ Python 17](#_Toc184583224)

[1.4.2. Thư viện sử dụng 17](#_Toc184583225)

[Chương 2. Xây dựng hệ thống nhận dạng đối tượng trong video 19](#_Toc184583226)

[2.1. Yêu cầu của bài toán 19](#_Toc184583227)

[2.2. Phương pháp tiếp cận 21](#_Toc184583228)

[2.3. Mô hình đề xuất 21](#_Toc184583229)

[2.3.1 Phát hiện đối tượng với YOLO 21](#_Toc184583230)

[2.3.2 Theo dõi đối tượng với DeepSORT 23](#_Toc184583231)

[2.4. Quy trình thực hiện 23](#_Toc184583232)

[2.4.1. Phát hiện vị trí và nhận diện đối tượng 24](#_Toc184583233)

[2.4.2. Theo dõi đối tượng qua các khung hình liên tiếp 25](#_Toc184583234)

[2.4.3. Xử lý các trường hợp góc nghiêng hoặc biến dạng 26](#_Toc184583235)

[2.4.4. Nhận diện và phân loại đối tượng 27](#_Toc184583236)

[2.4.5. Xuất kết quả và hiển thị trực quan 28](#_Toc184583237)

[Chương 3. Thực nghiệm Chương Trình 29](#_Toc184583238)

[3.1. Dữ liệu 29](#_Toc184583239)

[3.2. Các độ đo so sánh 30](#_Toc184583240)

[3.3. Kết quả 38](#_Toc184583241)

[KẾT LUẬN 44](#_Toc184583242)

[TÀI LIỆU THAM KHẢO 45](#_Toc184583243)

# DANH MỤC HÌNH ẢNH

[Hình 1.1: Thuật toán CNN 15](#_Toc184582492)

[Hình 1.2: Hình ảnh minh họa thuật toán CNN 15](#_Toc184582493)

[Hình 1.3: Thuật toán YOLO 16](#_Toc184582494)

[Hình 2.1: Hình ảnh mô tả bài toán 20](#_Toc184582495)

[Hình 2.2: Mô tả cách hoạt động của YOLO 22](#_Toc184582496)

[Hình 2.3: Minh hoạ cách bounding box hoạt động ở YOLO 23](#_Toc184582497)

[Hình 2.4: Mô tả các bước thực hiện dự án 24](#_Toc184582498)

[Hình 2.5: Xe đã được xác định bounding box 26](#_Toc184582499)

[Hình 2.6: Nhận diện và phân loại đối tượng 28](#_Toc184582500)

[Hình 3.1: Đường cong Precision-Confidence 32](#_Toc184582501)

[Hình 3.2: Đường cong Recall-Confidence 34](#_Toc184582502)

[Hình 3.3: Đường cong Precision-Recall 36](#_Toc184582503)

[Hình 3.4: Đường cong Precision-Recall thể hiện hiệu suất qua các mức confidence. 38](#_Toc184582504)

[Hình 3.5: Ma trận nhầm lẫn thể hiện hiệu suất phân loại của mô hình trên từng lớp dữ liệu. 41](#_Toc184582505)

[Hình 3.6: Đồ thị loss 42](#_Toc184582506)

[Hình 3.7: Các phương tiện được chương trình nhận diện thành công 43](#_Toc184582507)

[Hình 3.8: Một số người còn mờ không nhận diện được 43](#_Toc184582508)

# LỜI NÓI ĐẦU

Trong bối cảnh công nghệ ngày càng phát triển, các ứng dụng của Xử lý ảnh và Thị giác máy tính đang dần trở nên phổ biến trong nhiều lĩnh vực như an ninh, giám sát, y tế, giao thông và giải trí. Với mục tiêu cung cấp các giải pháp thông minh và tự động hóa, việc xây dựng hệ thống theo dõi đối tượng trong video đã và đang thu hút sự quan tâm lớn từ cộng đồng nghiên cứu cũng như ứng dụng thực tế.

Bài tập lớn này được thực hiện nhằm vận dụng các kiến thức lý thuyết và kỹ thuật đã học trong môn Xử lý ảnh và Thị giác máy tính để xây dựng một hệ thống theo dõi đối tượng trong video. Đề tài này không chỉ giúp sinh viên nắm vững các phương pháp cơ bản như xử lý khung hình, phát hiện đối tượng, và theo dõi chuyển động mà còn mở ra cơ hội tiếp cận với các thuật toán hiện đại như YOLO, CNN, hoặc DeepSORT.

Hệ thống được xây dựng sẽ bao gồm các tính năng như: phát hiện đối tượng trong các khung hình video, theo dõi các đối tượng qua các khung hình liên tiếp, và trực quan hóa kết quả theo dõi trên giao diện hiển thị. Các công cụ và ngôn ngữ lập trình như Python, OpenCV, TensorFlow sẽ được sử dụng để triển khai hệ thống.

Với đề tài "Xây dựng hệ thống theo dõi đối tượng trong video", nhóm chúng em hy vọng không chỉ hoàn thành mục tiêu của bài tập lớn mà còn tích lũy được các kỹ năng và kinh nghiệm thực tế để áp dụng trong các dự án thực tế sau này.

# LỜI CẢM ƠN

Trước tiên, nhóm chúng tôi xin gửi lời cảm ơn chân thành đến Bộ môn Xử lý ảnh và Thị giác máy tính, Khoa Công nghệ Thông tin, Trường Đại học Công Nghệ Đông Á, đã tạo điều kiện và cung cấp nền tảng kiến thức vững chắc để chúng tôi thực hiện bài tập lớn này.

Đặc biệt, chúng em xin bày tỏ lòng biết ơn sâu sắc tới cô Lương Thị Hồng Lan vì đã tận tình giảng dạy, định hướng, và hỗ trợ chúng tôi trong suốt quá trình học tập và triển khai đề tài. Sự hướng dẫn tận tình của thầy/cô không chỉ giúp nhóm hiểu sâu hơn về lý thuyết mà còn ứng dụng hiệu quả vào bài tập thực tế.

Chúng em cũng xin gửi lời cảm ơn đến các thành viên trong nhóm, vì sự phối hợp nhịp nhàng và tinh thần làm việc trách nhiệm đã góp phần giúp hoàn thành bài tập lớn này.

Mặc dù đã nỗ lực hết sức, nhưng bài tập lớn không thể tránh khỏi những thiếu sót. Chúng em mong nhận được sự góp ý từ thầy cô và các bạn để có thể hoàn thiện hơn trong các dự án tương lai.

Trân trọng cảm ơn!

# LỜI CAM ĐOAN

Chúng em, nhóm tác giả của bài tập lớn này, cam đoan rằng mọi nội dung được trình bày trong bài viết là hoàn toàn dựa trên nghiên cứu và hiểu biết của chúng tôi về chủ đề. Chúng em cam kết rằng không có phần nào của bài viết được sao chép hoặc tham khảo từ nguồn khác mà không được ghi rõ.

Chúng em cũng cam đoan rằng mọi thông tin, số liệu, và ý kiến cá nhân được trình bày trong bài viết là chân thực và không gian dối. Chúng em chịu trách nhiệm hoàn toàn về tính xác thực của nội dung đã được trình bày.

Cuối cùng, chúng em cam đoan rằng chúng tôi đã làm việc một cách tận tâm và nghiêm túc để hoàn thành bài tập lớn này với mục đích cung cấp thông tin hữu ích và chất lượng. Chúng em rất mong nhận được sự đánh giá và góp ý xây dựng từ các bạn đọc và giáo viên hướng dẫn để cải thiện và hoàn thiện bài viết.

**BẢNG PHÂN CÔNG CÔNG VIỆC**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên công việc** | **Công việc chia đến nhỏ nhất** | **Thành viên thực hiện** |
| 1 | Viết báo cáo | Word chương 1 | Nguyễn Thị Ánh |
| Word chương 2 | Bùi Thị Trang Ánh |
| Viết báo cáo chương 3 | Nguyễn Đắc Huy |
| Viết, sửa kiểm tra lại và hoàn thiện báo cáo chương 3 | Nguyễn Minh Thuận |
| Viết, chỉnh sửa lại, frommat lại bài trước khi in quyển | Nguyễn Thị Ánh |
| Viết kết luận sau khi đã hoàn thành | Bùi Thị Trang Ánh |
| 2 | Code | Thực hiện huấn luyện dữ liệu | Nguyễn Đắc Huy  Nguyễn Minh Thuận |
| Tiền xử lý, thu thập dữ liệu | Bùi Thị Trang Ánh  Nguyễn Thị Ánh |
| 3 | Slide báo cáo | thêm nội dung chi tiết vào slide | Nguyễn Thị Ánh |
| Hoàn thiện và chỉnh sửa | Bùi Thị Trang Ánh |

# Chương 1: Cơ sở lí thuyết

## 1.1. Nhận dạng và theo dõi đối tượng là gì?

Nhận dạng và theo dõi đối tượng là quá trình phát hiện, nhận diện và liên kết vị trí của đối tượng qua các khung hình trong video. Hệ thống cần xác định được các đối tượng, đồng thời duy trì danh tính của chúng khi di chuyển, xuất hiện hoặc biến mất khỏi khung hình.

Phát hiện đối tượng (Object Detection): Là bước xác định vị trí và loại đối tượng trong từng khung hình video. Các thuật toán phổ biến như YOLO và Faster R-CNN đảm bảo tốc độ xử lý nhanh và chính xác.

Theo dõi đối tượng (Object Tracking): Sau khi phát hiện, thuật toán theo dõi sẽ liên kết vị trí của đối tượng qua các khung hình liên tiếp. Các phương pháp như SORT và DeepSORT được sử dụng để gán ID duy nhất cho đối tượng, đảm bảo theo dõi liên tục.

## 1.2. Ứng dụng của hệ thống theo dõi đối tượng

### 1.2.1 Giám sát an ninh

Hệ thống theo dõi đối tượng trong video có thể đóng vai trò quan trọng trong việc giám sát an ninh, giúp theo dõi và nhận diện hành vi của người hoặc phương tiện di chuyển trong các khu vực có camera giám sát. Nhờ vào khả năng phân tích video theo thời gian thực, hệ thống có thể phát hiện các hành vi bất thường hoặc đáng ngờ, từ đó gửi cảnh báo kịp thời đến nhân viên bảo vệ hoặc cơ quan chức năng. Hệ thống còn có thể tự động nhận diện các đối tượng xâm nhập trái phép vào khu vực cấm hoặc các phương tiện di chuyển trái phép trong những khu vực hạn chế. Điều này giúp nâng cao khả năng bảo mật và giảm thiểu rủi ro trong các khu vực nhạy cảm như sân bay, ngân hàng, tòa nhà chính phủ hay các cơ sở quan trọng.

### 1.2.2 Hệ thống giao thông thông minh

Trong lĩnh vực giao thông, hệ thống theo dõi đối tượng có thể được ứng dụng để giám sát các phương tiện giao thông, phát hiện vi phạm và thu thập thông tin lộ trình. Ví dụ, hệ thống có thể theo dõi các phương tiện vượt đèn đỏ, đi sai làn hoặc chạy quá tốc độ. Bên cạnh đó, hệ thống cũng có thể giúp phân tích luồng giao thông và tối ưu hóa tín hiệu đèn giao thông nhằm giảm thiểu ùn tắc và cải thiện lưu thông. Đặc biệt, với các thành phố lớn, việc sử dụng hệ thống giám sát giao thông thông minh không chỉ giúp quản lý giao thông hiệu quả mà còn góp phần giảm thiểu tai nạn và tiết kiệm thời gian di chuyển cho người dân.

### 1.2.3 Quản lý đám đông

Khi sự kiện đông người diễn ra, chẳng hạn như các buổi hòa nhạc, lễ hội, sự kiện thể thao hoặc trong các trung tâm thương mại, việc theo dõi và quản lý đám đông là vô cùng quan trọng. Hệ thống theo dõi đối tượng có thể giám sát sự di chuyển của từng cá nhân trong đám đông, phát hiện những điểm tắc nghẽn hoặc sự di chuyển không bình thường có thể gây nguy hiểm. Các dấu hiệu bất thường, chẳng hạn như việc một nhóm người di chuyển theo một hướng mà không có mục đích rõ ràng hoặc sự tụ tập đột ngột ở một khu vực có thể dẫn đến nguy cơ chen lấn, đều có thể được hệ thống phát hiện và cảnh báo. Điều này giúp các nhân viên an ninh hoặc ban tổ chức có thể nhanh chóng hành động, tránh các tình huống nguy hiểm, bảo vệ an toàn cho mọi người tham gia.

### 1.2.4 Thể thao và giải trí

Trong ngành thể thao, hệ thống theo dõi đối tượng có thể được áp dụng để theo dõi vận động viên trong các trận đấu hoặc sự kiện thể thao. Hệ thống có thể phân tích các chuyển động, hành vi và chiến thuật của từng vận động viên, cung cấp dữ liệu giá trị cho huấn luyện viên, giúp họ đánh giá hiệu quả của chiến lược và đưa ra các quyết định điều chỉnh trong suốt trận đấu. Ngoài ra, công nghệ theo dõi đối tượng cũng có thể sử dụng trong quay phim và sản xuất các chương trình truyền hình thể thao, tự động theo dõi các đối tượng chính trong cảnh quay, giúp máy quay di chuyển linh hoạt theo đối tượng mà không cần can thiệp thủ công.

### 1.2.5 Quay phim và sản xuất truyền hình

Hệ thống theo dõi đối tượng còn có thể ứng dụng trong ngành quay phim và sản xuất truyền hình để tự động theo dõi các đối tượng trong cảnh quay. Điều này rất hữu ích trong việc quay các cảnh hành động, thể thao hay các chương trình truyền hình, giúp máy quay theo dõi và tập trung vào đối tượng chính mà không bị gián đoạn. Thậm chí, hệ thống còn có thể nhận diện được sự thay đổi trong hành động hoặc biểu cảm của đối tượng, từ đó điều chỉnh góc quay để tạo ra những cảnh quay chất lượng cao, nâng cao hiệu quả sản xuất và giảm thời gian hậu kỳ.

Hệ thống theo dõi đối tượng trong video không chỉ giúp cải thiện hiệu quả công việc và giải quyết nhiều vấn đề trong các lĩnh vực an ninh, giao thông, thể thao mà còn có ứng dụng rộng rãi trong các ngành công nghiệp giải trí và quản lý đám đông. Khi các công nghệ tiên tiến như trí tuệ nhân tạo và học sâu ngày càng phát triển, hệ thống này hứa hẹn sẽ trở thành một phần quan trọng trong các ứng dụng công nghệ trong tương lai.

## 1.3. Các kỹ thuật học máy sử dụng trong hệ thống

### 1.3.1. Mạng nơ-ron tích chập (Convolutional Neural Networks - CNN)

Ý tưởng: CNN sử dụng các lớp tích chập để tự động trích xuất các đặc trưng của đối tượng như hình dạng, màu sắc và kết cấu từ hình ảnh/video.

Ứng dụng: CNN là nền tảng cho nhiều phương pháp phát hiện đối tượng như Faster R-CNN, SSD, và YOLO.

Ưu điểm:

Hiệu quả trong trích xuất đặc trưng:

CNN có khả năng học và nhận diện các đặc trưng phức tạp từ hình ảnh, như đường nét, hình dạng, kết cấu và các chi tiết nhỏ. Điều này giúp tăng độ chính xác trong phát hiện đối tượng.

Khả năng mở rộng:

CNN có thể được mở rộng hoặc tinh chỉnh để áp dụng cho nhiều bài toán khác nhau, từ nhận diện đối tượng, phân loại hình ảnh, đến theo dõi và phát hiện bất thường.

Hỗ trợ từ cộng đồng:

Có nhiều framework phổ biến hỗ trợ xây dựng và triển khai CNN như TensorFlow, PyTorch, Keras, giúp giảm đáng kể thời gian phát triển.

Linh hoạt trong kết hợp:

CNN dễ dàng được tích hợp với các mô hình khác, chẳng hạn như RNN (Recurrent Neural Network) để xử lý dữ liệu tuần tự hoặc với các thuật toán theo dõi để duy trì danh tính đối tượng qua nhiều khung hình.

Nhược điểm:

Yêu cầu phần cứng cao:

CNN đòi hỏi tài nguyên tính toán lớn (GPU, TPU) để huấn luyện và suy luận, đặc biệt với dữ liệu video độ phân giải cao.

Tốc độ chậm:

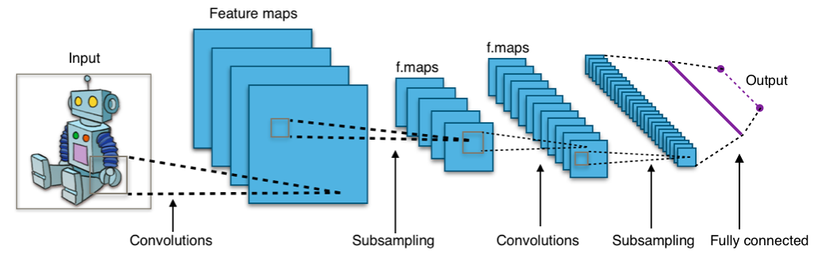
Phương pháp CNN thuần túy thường không phù hợp với các ứng dụng thời gian thực, vì cần nhiều bước xử lý để trích xuất đặc trưng và đưa ra dự đoán.

Cần dữ liệu lớn:

Để đạt được hiệu suất cao, CNN yêu cầu một lượng lớn dữ liệu được gắn nhãn chính xác. Việc thu thập và gắn nhãn dữ liệu này tốn nhiều thời gian và chi phí.

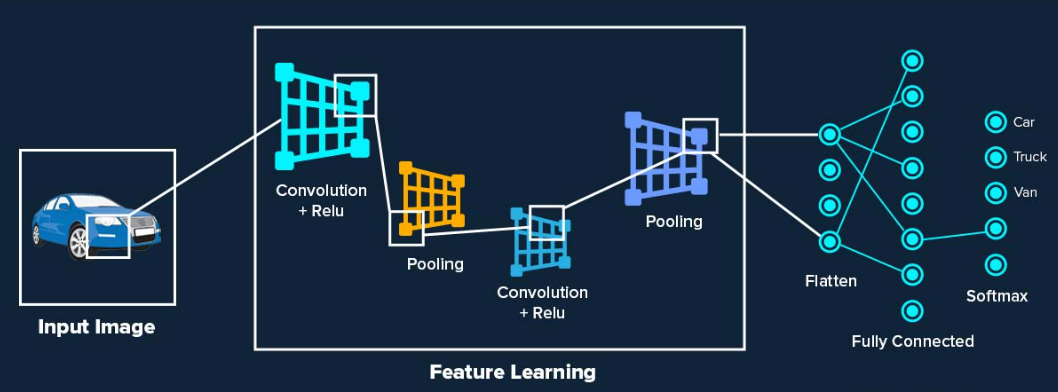
Không tối ưu cho phát hiện đối tượng nhanh:

CNN không phải là giải pháp lý tưởng khi cần phát hiện nhanh vị trí và số lượng nhiều đối tượng trong một hình ảnh hoặc video.



Hình 1.1: Thuật toán CNN

Hình ảnh minh họa CNN



Hình 1.2: Hình ảnh minh họa thuật toán CNN

### 1.3.2. YOLO (You Only Look Once)

Ý tưởng: YOLO chia hình ảnh thành các ô lưới và dự đoán bounding box cùng nhãn đối tượng trong một lần xử lý duy nhất.

Ứng dụng: Phát hiện đối tượng nhanh chóng và hiệu quả trong thời gian thực.

Ưu điểm:

Tốc độ cao:

YOLO được thiết kế tối ưu để xử lý trong thời gian thực. Nhờ đó, nó trở thành lựa chọn hàng đầu trong các ứng dụng yêu cầu tốc độ như giám sát giao thông, theo dõi an ninh, hoặc robot tự hành.

Khả năng phát hiện toàn diện:

YOLO chia hình ảnh thành các lưới nhỏ và dự đoán bounding box cho từng ô lưới. Điều này giúp phát hiện hiệu quả nhiều đối tượng cùng lúc, ngay cả khi chúng có kích thước hoặc vị trí khác nhau.

Độ chính xác cao trong nhận diện đối tượng:

Nhờ cách tiếp cận toàn cục (global), YOLO giảm thiểu lỗi phát hiện lặp và nhầm lẫn giữa các đối tượng.

Dễ dàng triển khai:

YOLO có nhiều phiên bản được tối ưu hóa, như YOLOv4, YOLOv5 hoặc YOLOv8, với các model được tinh chỉnh sẵn, dễ dàng áp dụng cho các bài toán thực tế.

Nhược điểm:

Hạn chế với các đối tượng nhỏ:

Trong một số trường hợp, YOLO gặp khó khăn khi phát hiện các đối tượng nhỏ hoặc khi chúng nằm gần biên của khung hình. Điều này xảy ra do quá trình chia lưới không đủ chi tiết.

Độ phức tạp trong điều chỉnh:

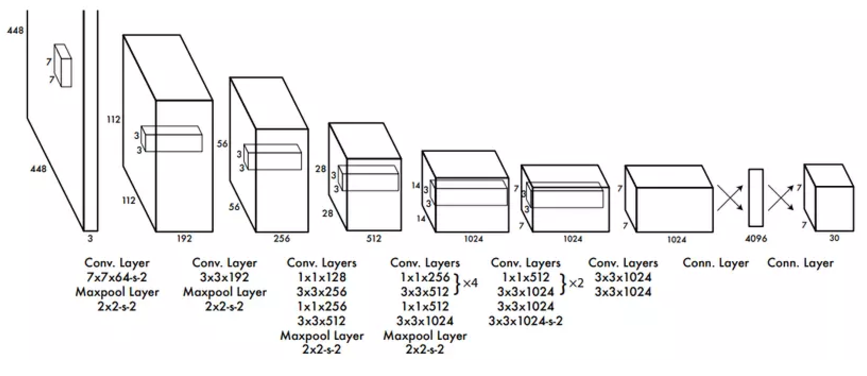
Để đạt hiệu suất tốt nhất, YOLO cần được tinh chỉnh hyperparameters cẩn thận. Điều này đòi hỏi hiểu biết sâu về thuật toán và cấu trúc của mô hình.

Hiệu suất giảm trong bối cảnh phức tạp:

YOLO có thể gặp khó khăn khi đối tượng bị che khuất, khi có quá nhiều đối tượng trong một khung hình, hoặc khi nền quá phức tạp.

Yêu cầu dữ liệu chất lượng cao:

Để tận dụng tối đa khả năng của YOLO, dữ liệu huấn luyện cần được gắn nhãn chính xác và phản ánh đúng các tình huống trong thực tế.



Hình 1.3: Thuật toán YOLO

### 1.3.3. SORT và DeepSORT

SORT (Simple Online and Realtime Tracking): Là thuật toán theo dõi thời gian thực, kết hợp phát hiện đối tượng với bộ lọc Kalman để dự đoán vị trí tiếp theo của đối tượng.

DeepSORT: Là phiên bản cải tiến của SORT, kết hợp thêm đặc trưng từ CNN để tăng độ chính xác trong việc duy trì danh tính đối tượng.

## 1.4. Ngôn ngữ lập trình và các thư viện

### 1.4.1. Ngôn ngữ Python

Python là một ngôn ngữ lập trình được sử dụng rộng rãi trong các ứng dụng web, phát triển phần mềm, khoa học dữ liệu và máy học (ML). Python có một hệ sinh thái phong phú các thư viện, giúp nó trở thành ngôn ngữ phổ biến trong các lĩnh vực như phát triển web, khoa học dữ liệu, trí tuệ nhân tạo, và thị giác máy tính.

Ưu điểm:

Dễ học và dễ sử dụng: Python có cú pháp rõ ràng, dễ đọc, phù hợp cho cả người mới bắt đầu và lập trình viên chuyên nghiệp.

Thư viện phong phú: Python cung cấp hàng nghìn thư viện như NumPy, Pandas, TensorFlow, OpenCV phục vụ đa dạng các lĩnh vực.

Tương thích cao: Chạy trên nhiều nền tảng khác nhau mà không cần sửa đổi mã nguồn.

Cộng đồng mạnh mẽ: Python có một cộng đồng người dùng rộng lớn, giúp dễ dàng tìm kiếm tài liệu, hỗ trợ, và ví dụ.

Hỗ trợ tốt cho AI/ML: Là ngôn ngữ phổ biến nhất trong lĩnh vực trí tuệ nhân tạo và học máy nhờ khả năng tích hợp các thư viện chuyên dụng.

Nhược điểm:

Tốc độ chậm hơn: Python là ngôn ngữ thông dịch nên tốc độ xử lý thường chậm hơn so với C/C++ hoặc Java.

Không phù hợp cho ứng dụng thời gian thực yêu cầu hiệu suất cao: Python không phải là lựa chọn tối ưu cho các ứng dụng yêu cầu xử lý siêu nhanh.

Hạn chế trong đa luồng: Do GIL (Global Interpreter Lock), Python không thể tận dụng hoàn toàn các CPU đa nhân trong nhiều trường hợp.

### 1.4.2. Thư viện sử dụng

1. OpenCV ( Open Source Computer Vision Library):

Là thư viện mã nguồn mở chuyên về xử lý hình ảnh và video

Cung cấp nhiều chức năng như:

Đọc, ghi và hiển thị hình ảnh/video

Tiền xử lý ảnh (lọc, điều chỉnh độ sáng, làm mờ)

Phát hiện và nhận diện đối tượng

Trích xuất đặc trưng từ hình ảnh

Hỗ trợ các thuật toán thị giác máy tính

1. TensorFlow/Keras:

TensorFlow: Thư viện học máy và học sâu của Google

Keras: API bậc cao để xây dựng mạng neural

Chức năng chính:

Xây dựng các mô hình mạng neural như CNN (Convolutional Neural Network)

Huấn luyện mô hình YOLO (You Only Look Once) cho phát hiện đối tượng

Hỗ trợ quá trình huấn luyện và đánh giá mô hình

1. NumPy:

Thư viện tính toán khoa học cho Python

Chuyên xử lý mảng và ma trận

Cung cấp các hàm toán học hiệu quả

Tối ưu cho các phép tính số học phức tạp

1. Pandas:

Thư viện phân tích dữ liệu

Quản lý và xử lý dữ liệu theo bảng

Hỗ trợ đọc/ghi nhiều định dạng

Xử lý dữ liệu thiếu, thống kê

1. DeepSORT (Deep Simple Online and Realtime Tracking):

Thuật toán theo dõi đối tượng tiên tiến

Kết hợp với các mô hình phát hiện đối tượng

Theo dõi liên tục và chính xác các đối tượng qua các khung hình

Ứng dụng trong giám sát, an ninh, xe tự hành

# Chương 2. Xây dựng hệ thống nhận dạng đối tượng trong video

## 2.1. Yêu cầu của bài toán

Theo dõi đối tượng trong video là một bài toán quan trọng trong lĩnh vực thị giác máy tính và học sâu, với các ứng dụng trong giám sát an ninh, phân tích hành vi, phương tiện tự hành, và tương tác người-máy. Mục tiêu của bài toán là phát hiện, nhận dạng, và theo dõi liên tục vị trí của một hoặc nhiều đối tượng trong chuỗi khung hình video.

Hệ thống này phải đối mặt với nhiều thách thức, bao gồm:

Che khuất và thay đổi môi trường: Các đối tượng có thể bị che khuất một phần hoặc toàn bộ trong một số khung hình, hoặc di chuyển qua các khu vực có ánh sáng, màu sắc, và nền khác nhau.

Thay đổi hình dáng và kích thước: Do góc nhìn, khoảng cách, hoặc chuyển động, hình dạng và kích thước của đối tượng có thể thay đổi theo thời gian.

Nhiễu và chất lượng video thấp: Nhiễu tín hiệu hoặc độ phân giải thấp có thể ảnh hưởng đến độ chính xác của việc phát hiện và theo dõi.

Số lượng đối tượng nhiều: Khi cần theo dõi nhiều đối tượng đồng thời, việc duy trì nhận dạng chính xác từng đối tượng trở nên phức tạp.

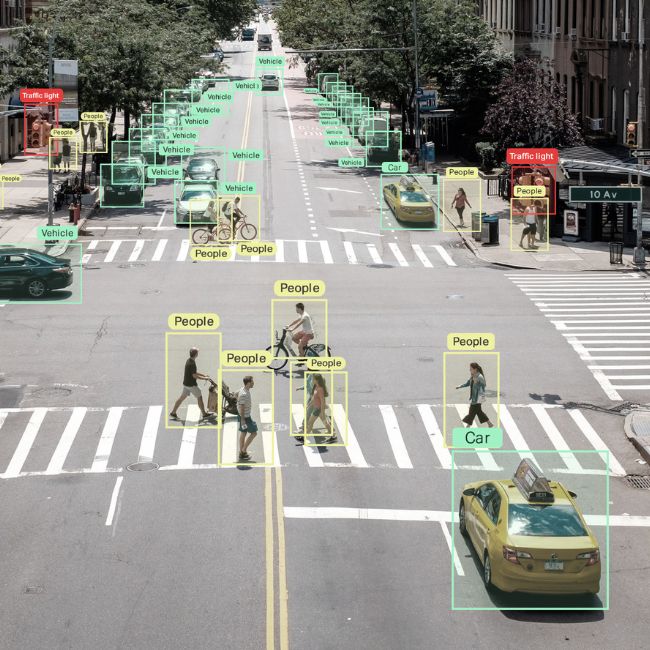
Hệ thống thường bao gồm các thành phần chính:

Phát hiện đối tượng (Object Detection): Sử dụng các mô hình như YOLO, SSD, hoặc Faster R-CNN để nhận diện đối tượng trong từng khung hình.

Theo dõi đối tượng (Object Tracking): Áp dụng các thuật toán như Kalman Filter, SORT, hoặc DeepSORT để dự đoán và theo dõi vị trí của đối tượng qua các khung hình liên tiếp.

Liên kết khung hình (Data Association): Đảm bảo mỗi đối tượng được gán một ID duy nhất và giữ nguyên qua các khung hình.

Tích hợp trí tuệ nhân tạo (AI): Các mạng nơ-ron tích chập (CNN) hoặc mạng tái phát (RNN) có thể được sử dụng để nâng cao khả năng theo dõi và dự đoán hành vi của đối tượng.



Hình 2.1: Hình ảnh mô tả bài toán

Các yêu cầu cụ thể của hệ thống bao gồm:

Khả năng phát hiện chính xác:

Hệ thống cần nhận diện chính xác đối tượng mục tiêu ngay cả khi kích thước nhỏ, có sự thay đổi góc nhìn, hoặc bị che khuất một phần.

Duy trì theo dõi liên tục:

Theo dõi xuyên suốt dù đối tượng di chuyển nhanh, bị che khuất, hoặc quay trở lại sau khi ra khỏi khung hình.

Khả năng hoạt động trong môi trường thay đổi:

Đảm bảo độ chính xác ngay cả khi điều kiện ánh sáng, màu nền, hoặc góc nhìn thay đổi.

Tính năng bổ sung:

Hiển thị trực quan các vùng đối tượng trên giao diện người dùng.

Hỗ trợ theo dõi đồng thời nhiều đối tượng.

Cảnh báo hoặc gửi thông báo khi có hành vi bất thường.

## 2.2. Phương pháp tiếp cận

Hệ thống được xây dựng với các bước sau:

Phát hiện đối tượng (Object Detection):

Áp dụng mô hình YOLO để xác định vị trí và phân loại đối tượng trong từng khung hình.

Theo dõi đối tượng (Object Tracking):

Sử dụng các thuật toán như Kalman Filter hoặc DeepSORT để dự đoán và theo dõi vị trí qua các khung hình.

Tích hợp dữ liệu khung hình (Data Association):

Gán ID duy nhất cho mỗi đối tượng, duy trì nhận dạng qua các khung hình liên tiếp.

## 2.3. Mô hình đề xuất

### 2.3.1 Phát hiện đối tượng với YOLO

YOLO (You Only Look Once) là mô hình phát hiện đối tượng tiên tiến, dựa trên mạng nơ-ron tích chập (CNN). YOLO có khả năng:

Nhận diện nhanh và chính xác nhiều đối tượng trong thời gian thực.

Chia ảnh thành lưới (grid), dự đoán bounding boxes cùng nhãn lớp (class label) cho từng đối tượng.

Quy trình hoạt động:

Tiền xử lý hình ảnh:

Ảnh được chuẩn hóa về kích thước cố định, ví dụ 416x416 pixels.

Trích xuất đặc trưng:

Sử dụng các lớp tích chập để nhận diện hình dáng, màu sắc, và cấu trúc đối tượng.

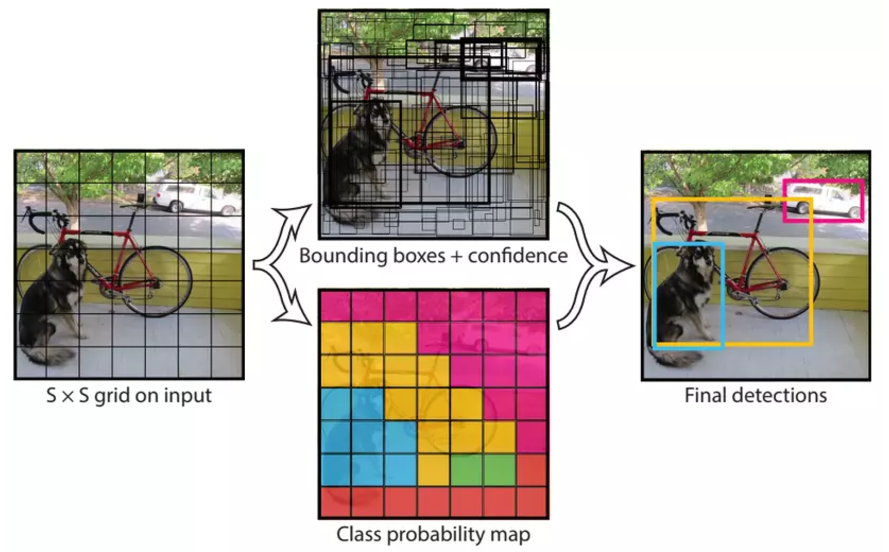
Dự đoán bounding boxes:

Mỗi ô lưới dự đoán tọa độ, kích thước, điểm tin cậy, và nhãn lớp.

Hậu xử lý:

Áp dụng Non-Maximum Suppression (NMS) để loại bỏ bounding boxes trùng lặp, giữ lại những box tốt nhất.

Hình ảnh minh họa YOLO



Hình 2.2: Mô tả cách hoạt động của YOLO

Đầu vào của một mô hình là một ảnh, mô hình sẽ nhận dạng ảnh đó có đối tượng nào hay không, sau đó sẽ xác định toạ độ của đối tượng trong bức ảnh. Ảnh đầu vào được chia thành S x S ô, thường thì sẽ là 3x3, 7x7… Việc chia ô này có ảnh hưởng tới việc mô hình phát hiện đối tượng, và đó chính là lí do em lựa chọn YOLO để hoàn thành dự án lần này (với bounding box (so với v3) và 2 kĩ thuật mới được thêm từ YOLO v4 là Mosaic Augmentation và Self Adversarial Training (SAT) thì ta sẽ có hình minh hoạ cách hoạt động của nó ở bên dưới).



Hình 2.3: Minh hoạ cách bounding box hoạt động ở YOLO

### 2.3.2 Theo dõi đối tượng với DeepSORT

DeepSORT cải thiện khả năng theo dõi bằng cách:

Kết hợp thông tin hình ảnh (appearance) và quỹ đạo (trajectory).

Duy trì liên tục ID đối tượng ngay cả khi di chuyển nhanh hoặc bị che khuất.

## 2.4. Quy trình thực hiện

Hiện nay, trên thế giới đã có rất nhiều cách tiếp cận khác nhau trong việc theo dõi đối tượng trong video. Tuy nhiên, trong phạm vi bài báo cáo này, hệ thống sẽ giải quyết vấn đề theo 3 bước chính:

Phát hiện vị trí và nhận diện đối tượng trong từng khung hình:

Sử dụng mô hình YOLO để phát hiện đối tượng trong từng khung hình video.

Tạo bounding boxes (khung giới hạn) để đánh dấu các vùng chứa đối tượng được phát hiện.

Thông tin bao gồm tọa độ, kích thước, và nhãn của từng đối tượng.

Gán ID và theo dõi đối tượng qua các khung hình liên tiếp:

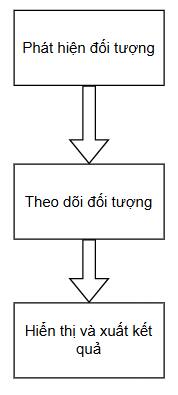
Áp dụng thuật toán DeepSORT để liên kết và theo dõi đối tượng xuyên suốt video.

Mỗi đối tượng được gán một ID duy nhất và được duy trì khi di chuyển giữa các khung hình.

Dự đoán vị trí tiếp theo của đối tượng dựa trên thông tin lịch sử và quỹ đạo di chuyển.

Xuất kết quả theo dõi và hiển thị trực quan:

Hiển thị bounding boxes và ID của đối tượng theo thời gian thực trên video.



Hình 2.4: Mô tả các bước thực hiện dự án

### 2.4.1. Phát hiện vị trí và nhận diện đối tượng

YOLO là một mô hình phát hiện đối tượng mạnh mẽ, được huấn luyện trước trên các tập dữ liệu lớn và đa dạng như COCO. Mô hình này có khả năng phát hiện chính xác các đối tượng trong khung hình với tốc độ cao, rất phù hợp cho các ứng dụng thời gian thực.

Các bước thực hiện:

Bước 1: Chuẩn hóa kích thước khung hình video về 640x640 pixels để phù hợp với đầu vào của mô hình YOLO.

Bước 2: YOLO quét qua từng khung hình video và phát hiện các vùng chứa đối tượng thông qua bounding boxes.

Bước 3: Mỗi bounding box sẽ được gán một nhãn lớp (class) và độ tin cậy (confidence score) để xác định đối tượng cụ thể (ví dụ: người, phương tiện, động vật).

Bước 4: Áp dụng Non-Maximum Suppression (NMS) để loại bỏ các bounding boxes trùng lặp và chỉ giữ lại các box có độ tin cậy cao nhất.

Bước 5: Lưu trữ thông tin bounding boxes gồm tọa độ (xmin, ymin, xmax, ymax), nhãn lớp, và điểm tin cậy vào một danh sách để phục vụ bước theo dõi tiếp theo.

### 2.4.2. Theo dõi đối tượng qua các khung hình liên tiếp

Sau khi phát hiện được các đối tượng, hệ thống sẽ tiến hành theo dõi các đối tượng đó xuyên suốt các khung hình trong video.

Các bước thực hiện:

Bước 1: Gán ID duy nhất cho từng đối tượng tại khung hình đầu tiên để theo dõi chúng qua các khung hình tiếp theo.

Bước 2: Sử dụng thuật toán DeepSORT để dự đoán vị trí tiếp theo của đối tượng dựa trên:

Vị trí hiện tại của bounding box.

Vận tốc và quỹ đạo di chuyển của đối tượng.

Bước 3: Liên kết đối tượng trong khung hình hiện tại với khung hình trước đó bằng cách tính khoảng cách giữa các bounding boxes thông qua thuật toán Hungarian.

Bước 4: Xử lý trường hợp che khuất hoặc mất dấu tạm thời:

Nếu đối tượng bị che khuất trong vài khung hình, hệ thống sẽ dự đoán vị trí của đối tượng và tiếp tục theo dõi khi nó xuất hiện trở lại.

Nếu đối tượng không xuất hiện sau một số khung hình nhất định (ví dụ: 10 khung), hệ thống sẽ ngừng theo dõi ID đó.



Hình 2.5: Xe đã được xác định bounding box

### 2.4.3. Xử lý các trường hợp góc nghiêng hoặc biến dạng

Trong quá trình theo dõi, một số đối tượng có thể xuất hiện với góc nghiêng hoặc biến dạng nhẹ do chuyển động. Để đảm bảo độ chính xác, hệ thống thực hiện các bước sau:

Bước 1: Làm mờ ảnh bằng bộ lọc Gaussian để loại bỏ nhiễu và giữ lại các đường biên sắc nét.

Bước 2: Sử dụng Canny Edge Detection để phát hiện các đường biên của đối tượng.

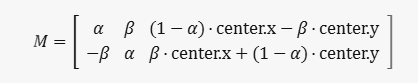
Bước 3: Áp dụng Hough Transform để xác định các đường thẳng đặc trưng trong bounding box.

Bước 4: Tính toán góc nghiêng của đối tượng so với trục ngang (Ox) bằng công thức:



Bước 5: Sử dụng hàm cv2.getRotationMatrix2D() trong OpenCV để xoay lại bounding box về phương thẳng đứng.

Hàm này sử dụng công thức:



Trong đó

α = scale ⋅ cos(angle)

β = scale ⋅ sin(angle)

Với các tham số:

Center: toạ độ của tâm xoay (x, y).

Angle: Góc xoay tính bằng độ (vừa được tính toán với công thức ở trên).

Scale: Hệ số tỉ lệ (xác định mức độ phóng to thu nhỏ của hình ảnh khi xoay).

### 2.4.4. Nhận diện và phân loại đối tượng

Sau khi phát hiện và theo dõi đối tượng, hệ thống sẽ nhận diện nhãn lớp của từng đối tượng và phân loại chúng.

Các bước thực hiện:

Bước 1: Bounding box của đối tượng được đưa vào mô hình YOLO.

Bước 2: Lưu trữ thông tin nhãn lớp, tọa độ và ID của đối tượng.

Bước 3: Hiển thị nhãn lớp và ID trực tiếp trên video (ví dụ: “Person\_1”, “Car\_3”).

Bước 4: Tính toán độ tin cậy của nhận diện

Bước 5: Lọc kết quả theo ngưỡng tin cậy



Hình 2.6: Nhận diện và phân loại đối tượng

### 2.4.5. Xuất kết quả và hiển thị trực quan

Cuối cùng, hệ thống sẽ hiển thị kết quả theo dõi.

Hiển thị trực quan:

Các bounding boxes và ID của đối tượng được vẽ trên video theo thời gian thực.

Quỹ đạo di chuyển của đối tượng được hiển thị dưới dạng đường nối trên video.

Xuất kết quả:

Xuất video kết quả với bounding boxes và ID đã được gắn.

# Chương 3. Thực nghiệm Chương Trình

## 3.1. Dữ liệu

### ***3.1.1 Dữ liệu hình ảnh***

Hệ thống sử dụng dữ liệu được thu thập từ Roboflow, một nền tảng chuyên biệt cung cấp và hỗ trợ xử lý các tập dữ liệu dành cho các bài toán thị giác máy tính. Đây là một công cụ phổ biến và hiệu quả, được nhiều nhà phát triển tin dùng nhờ khả năng tối ưu hóa quá trình chuẩn bị dữ liệu.

Roboflow được lựa chọn vì nền tảng này cung cấp các tập dữ liệu đã được chú thích sẵn với chất lượng cao, giúp giảm đáng kể thời gian cần thiết để chuẩn bị dữ liệu cho các mô hình học máy. Ngoài ra, Roboflow còn hỗ trợ khả năng chuyển đổi định dạng dữ liệu, phù hợp với nhiều mô hình khác nhau như YOLO, Pascal VOC, COCO, đảm bảo sự linh hoạt và dễ dàng tích hợp vào quy trình phát triển.

Mô tả dữ liệu

Tập dữ liệu sử dụng trong hệ thống bao gồm các thông tin cụ thể như sau:

Số lượng ảnh: 5,827 ảnh. Đây là số lượng đủ lớn để đảm bảo mô hình có thể học và nhận diện được các đặc trưng khác nhau trong bài toán theo dõi đối tượng.

Các lớp đối tượng:

person: Nhận diện người.

car: Nhận diện xe hơi.

vehicle: Nhận diện phương tiện giao thông khác.

Các lớp này được lựa chọn dựa trên tính ứng dụng thực tiễn của hệ thống, phù hợp với các tình huống như giám sát giao thông hoặc theo dõi đối tượng trong môi trường đô thị.

### ***3.1.2 Định dạng dữ liệu:***

Dữ liệu được lưu trữ dưới dạng tệp YOLO .txt để cung cấp thông tin về bounding box, kèm theo hình ảnh ở định dạng .jpg hoặc .png.

Các định dạng này được chuẩn hóa và tối ưu để phù hợp với hệ thống YOLO, đảm bảo tốc độ xử lý nhanh và độ chính xác cao.

Tỷ lệ phân chia dữ liệu:

Tập huấn luyện (train): 85% dữ liệu.

Tập kiểm thử (test): 9% dữ liệu.

Tập xác thực (validation): 5% dữ liệu.

Tỷ lệ này đảm bảo rằng mô hình có đủ dữ liệu để học, đồng thời duy trì được tính khách quan và độ chính xác khi đánh giá hiệu suất.

Tiền xử lý dữ liệu

Trước khi đưa dữ liệu vào mô hình, các bước tiền xử lý được thực hiện để tối ưu hóa khả năng học của hệ thống, bao gồm:

Chuyển đổi kích thước ảnh:

Tất cả các ảnh đầu vào được chuẩn hóa về kích thước 320x320 pixels, đây là kích thước tiêu chuẩn giúp tối ưu hóa hiệu suất của YOLO.

Quá trình này đảm bảo các ảnh có cùng độ phân giải, giúp giảm tải xử lý và tăng tính nhất quán trong quá trình huấn luyện.

Chuẩn hóa màu sắc:

Các ảnh được chuẩn hóa giá trị pixel để phù hợp với các giá trị đầu vào của mạng nơ-ron tích chập (CNN).

Tăng cường dữ liệu (Data Augmentation):

Trong một số trường hợp, các kỹ thuật như xoay ảnh, thay đổi độ sáng hoặc cắt ảnh có thể được sử dụng để tạo ra nhiều biến thể khác nhau của cùng một hình ảnh. Điều này giúp cải thiện khả năng tổng quát hóa của mô hình.

Loại bỏ ảnh nhiễu:

Các ảnh bị lỗi hoặc có chất lượng kém (nhòe, mờ) được lọc ra để đảm bảo tập dữ liệu đầu vào có chất lượng cao.

## 3.2. Các độ đo so sánh

### ***3.2.1 Precision ( độ chính xác)***

Đánh giá hiệu quả của mô hình thông qua các độ đo

Để đánh giá chất lượng và hiệu quả của mô hình sau quá trình huấn luyện, nghiên cứu đã sử dụng một số chỉ số phổ biến, trong đó \*\*Precision (P)\*\* là một trong những độ đo quan trọng nhất. Precision được sử dụng để đo lường mức độ chính xác của các dự đoán, tức là tỷ lệ giữa số lượng dự đoán đúng trên tổng số dự đoán mà mô hình thực hiện. Dưới đây là chi tiết các kết quả precision theo từng lớp đối tượng:

Precision tổng thể: 0.732 (73.2%)

Precision tổng thể là chỉ số tổng quát, phản ánh khả năng dự đoán chính xác của mô hình trên tất cả các lớp đối tượng trong tập dữ liệu. Với giá trị 73.2%, mô hình đạt độ chính xác cao và có khả năng nhận diện các đối tượng một cách đáng tin cậy.

Precision tổng thể cho thấy rằng trung bình, hơn 7/10 dự đoán của mô hình là chính xác. Điều này là một kết quả khả quan, đặc biệt trong các bài toán thị giác máy tính phức tạp như phát hiện và theo dõi đối tượng.

Precision theo từng lớp đối tượng

Car: 0.655 (65.5%)

Đây là giá trị precision đạt được đối với lớp đối tượng "Car".

Mặc dù độ chính xác không cao bằng các lớp khác, nhưng vẫn duy trì ở mức chấp nhận được. Điều này có thể do các yếu tố như đa dạng về kích thước, màu sắc và góc nhìn của xe trong tập dữ liệu.

Person: 0.877 (87.7%)

Lớp "Person" có độ chính xác cao nhất trong các lớp đối tượng, đạt 87.7%.

Kết quả này phản ánh rằng mô hình có khả năng nhận diện người trong video rất tốt, ngay cả khi gặp các điều kiện môi trường khác nhau. Điều này có thể là do sự đồng nhất cao của đặc trưng "Person" trong tập dữ liệu huấn luyện.

Vehicle: 0.665 (66.5%

Đối với lớp "Vehicle", mô hình đạt precision ở mức 66.5%.

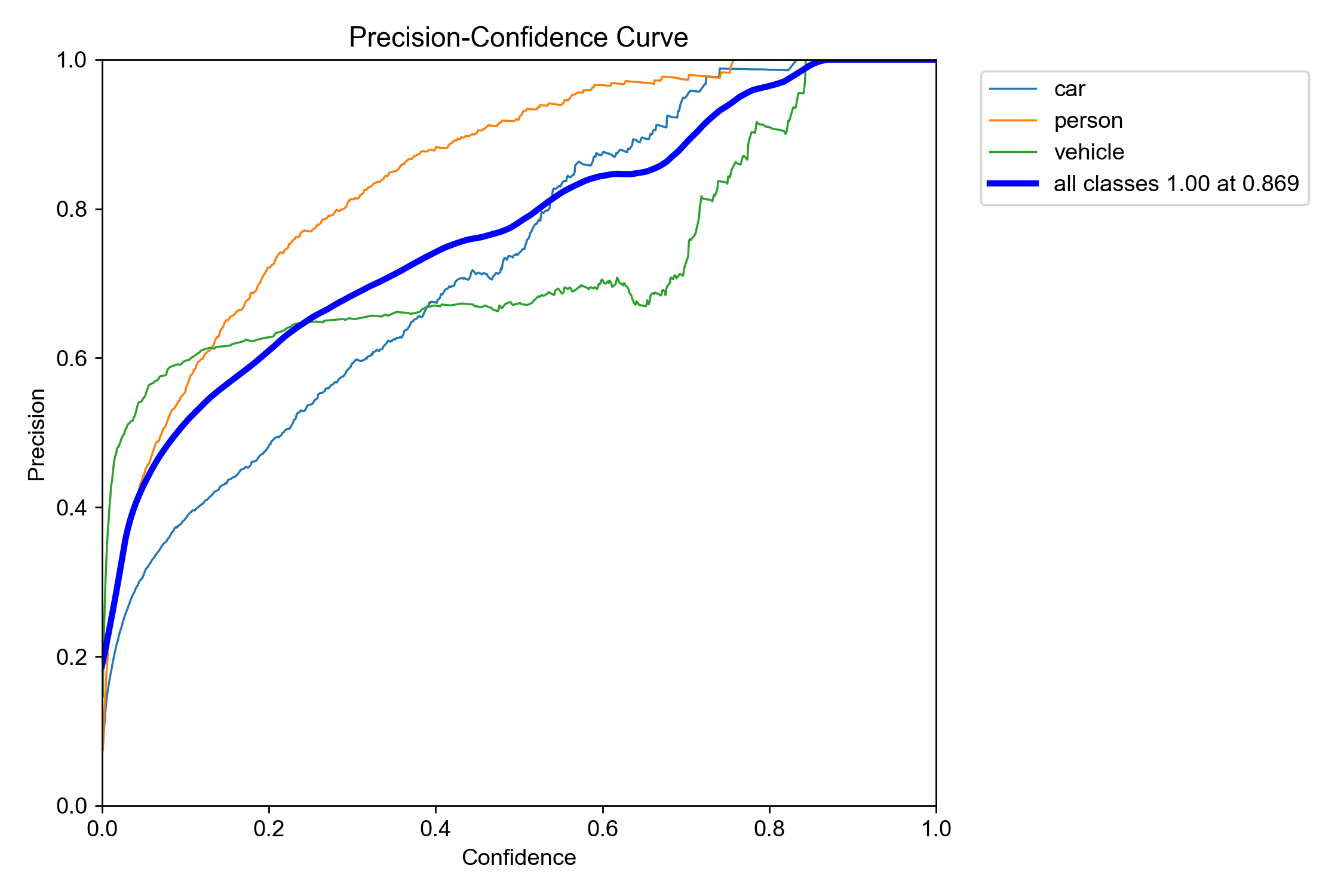
Giá trị này tương đương với lớp "Car" và thể hiện rằng mô hình vẫn còn gặp khó khăn khi xử lý các đối tượng thuộc lớp "Vehicle", có thể do sự chồng chéo hoặc tương đồng về đặc trưng với lớp "Car".

Tổng hợp và nhận xét

Precision cao nhất thuộc về lớp "Person" (87.7%), cho thấy khả năng nhận diện con người vượt trội so với các đối tượng khác.

Precision của các lớp "Car" (65.5%) và "Vehicle" (66.5%) có thể được cải thiện hơn nữa thông qua việc tinh chỉnh dữ liệu huấn luyện, bổ sung thêm ảnh mẫu hoặc tăng cường độ đa dạng về góc nhìn và điều kiện môi trường.

Precision tổng thể (73.2%) là một chỉ số tốt, chứng minh rằng mô hình có hiệu suất đáng tin cậy và phù hợp để áp dụng vào các bài toán thực tế, đặc biệt trong việc phát hiện và theo dõi đối tượng trong video.

Kết quả đánh giá hiệu quả dựa trên precision đã cung cấp cái nhìn tổng quát về khả năng của mô hình. Với precision trung bình trên 70%, mô hình được xem là hoạt động tốt, đặc biệt trong các bài toán nhận diện đa đối tượng phức tạp. Tuy nhiên, vẫn cần tập trung vào việc cải thiện độ chính xác cho các lớp như "Car" và "Vehicle" để nâng cao hiệu suất tổng thể. 

Hình 3.1: Đường cong Precision-Confidence

### ***3.2.2 Recall ( độ nhạy )***

#### *a. Đánh giá hiệu quả của mô hình thông qua Recall (R)*

Để đánh giá khả năng phát hiện đầy đủ các đối tượng trong video, chỉ số Recall (R) được sử dụng. Recall, hay còn gọi là độ nhạy, đo lường khả năng của mô hình trong việc phát hiện tất cả các đối tượng thuộc lớp mục tiêu trong tập dữ liệu. Chỉ số này thể hiện tỷ lệ giữa số lượng đối tượng được phát hiện đúng với tổng số đối tượng thực tế. Dưới đây là chi tiết các kết quả recall theo từng lớp đối tượng và đánh giá tổng thể:

Recall tổng thể: 0.763 (76.3%)

Recall tổng thể là chỉ số tổng hợp, phản ánh khả năng mô hình phát hiện tất cả các đối tượng trên toàn bộ tập dữ liệu. Với giá trị đạt 76.3%, mô hình thể hiện khả năng phát hiện tốt, bao quát hầu hết các đối tượng trong các khung hình video.

Đây là một kết quả đáng khích lệ, đặc biệt trong các bài toán phát hiện đối tượng phức tạp như video trực tuyến. Điều này cho thấy mô hình có khả năng phát hiện các đối tượng một cách hiệu quả mà ít bỏ sót.

Recall theo từng lớp đối tượng

Car: 0.764 (76.4%)

Lớp "Car" đạt recall gần với mức tổng thể (76.4%).

Kết quả này thể hiện rằng mô hình có khả năng phát hiện tốt các phương tiện di chuyển như ô tô, với tỷ lệ bỏ sót đối tượng ở mức thấp.

Person: 0.598 (59.8%)

Đây là lớp có recall thấp nhất trong các lớp đối tượng, đạt 59.8%.

Giá trị này cho thấy mô hình vẫn gặp khó khăn trong việc phát hiện tất cả các đối tượng "Person", đặc biệt trong các tình huống phức tạp như người bị che khuất hoặc xuất hiện trong điều kiện ánh sáng kém.

Vehicle: 0.926 (92.6%)

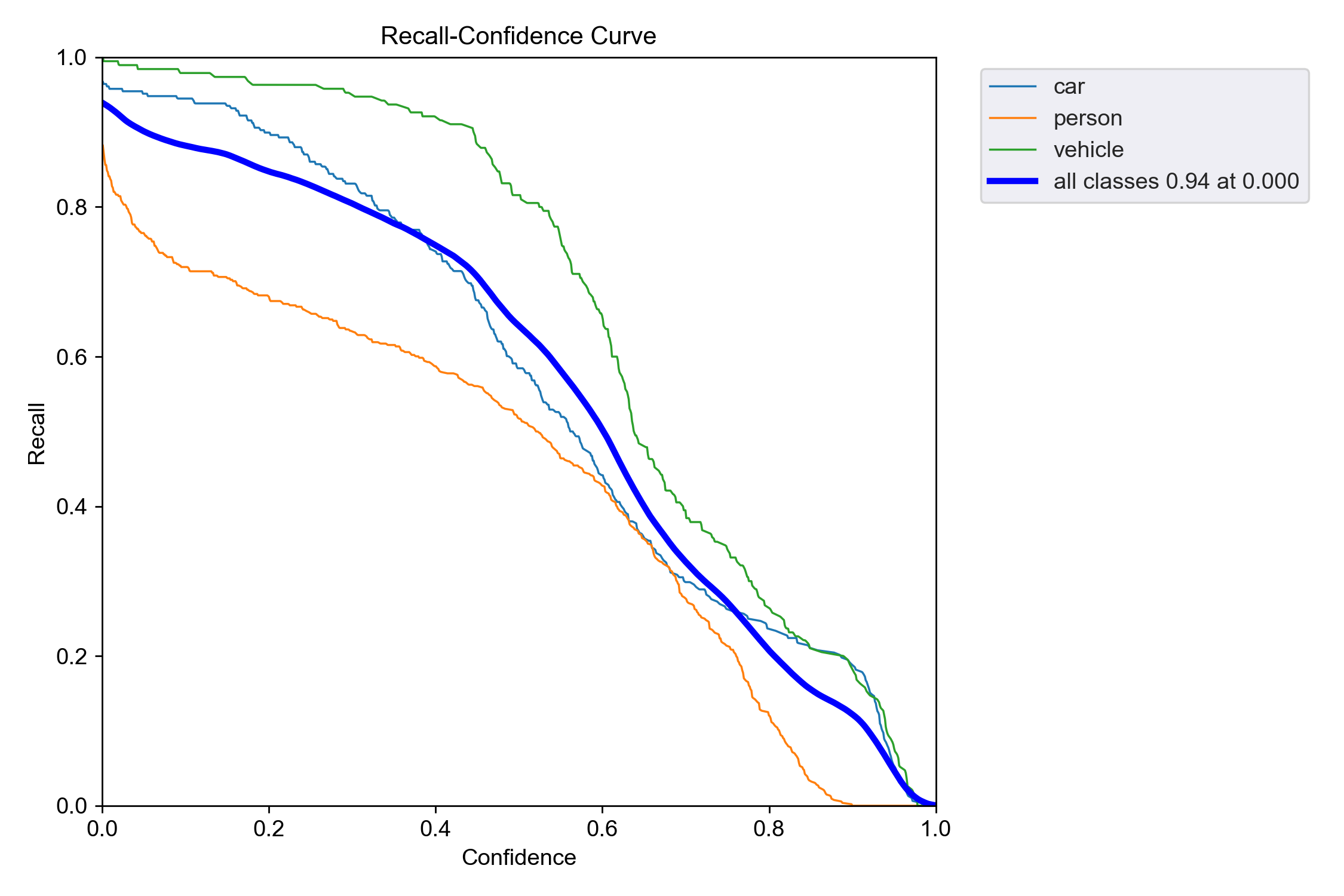
Lớp "Vehicle" đạt recall cao nhất với 92.6%.

Đây là một kết quả rất tốt, cho thấy mô hình gần như không bỏ sót bất kỳ phương tiện nào thuộc lớp này. Điều này có thể nhờ vào sự khác biệt rõ ràng về đặc trưng của các phương tiện trong tập dữ liệu huấn luyện.

Recall của lớp "Vehicle" (92.6%) nổi bật hơn hẳn, cho thấy mô hình hoạt động xuất sắc trong việc phát hiện các đối tượng thuộc lớp này.

Mặc dù lớp "Car" đạt recall ở mức tốt (76.4%), vẫn có khoảng cách so với lớp "Vehicle". Điều này có thể do sự chồng chéo giữa các lớp hoặc sự đa dạng chưa đủ trong tập huấn luyện.

Lớp "Person" có recall thấp nhất (59.8%), cho thấy rằng mô hình cần được cải thiện khả năng phát hiện người trong các tình huống phức tạp. Điều này có thể được khắc phục bằng cách bổ sung thêm dữ liệu mẫu đa dạng hơn hoặc cải tiến kỹ thuật tiền xử lý dữ liệu.

Với recall tổng thể đạt 76.3%, mô hình được đánh giá là hoạt động tốt, đặc biệt trong việc phát hiện các đối tượng thuộc lớp "Vehicle". Tuy nhiên, để tăng hiệu quả trong các ứng dụng thực tế, cần chú trọng cải thiện recall của lớp "Person", nhằm đảm bảo rằng mô hình có thể nhận diện được hầu hết các cá thể con người trong video. Việc này có thể bao gồm thu thập thêm dữ liệu huấn luyện và tối ưu hóa các tham số mô hình. 

Hình 3.2: Đường cong Recall-Confidence

#### *b .Đánh giá mAP50 (IoU=0.5)*

mAP50 tổng thể: 0.763 (76.3%)

Mô hình đạt độ chính xác khá tốt với tổng thể mAP50 đạt 76.3%. Điều này cho thấy mô hình có khả năng nhận diện và phân loại các đối tượng trong ảnh với mức độ chính xác khá cao khi sử dụng ngưỡng Intersection over Union (IoU) = 0.5. Ngưỡng này được sử dụng để đo độ chồng lấp giữa bounding box dự đoán và bounding box ground truth, với giá trị cao hơn thể hiện độ chính xác của dự đoán tốt hơn.

Đánh giá theo từng lớp đối tượng:

Car: 0.775 (77.5%)

Đối với lớp "Car", mô hình có mAP50 đạt 77.5%, cho thấy hiệu suất khá tốt trong việc nhận diện ô tô. Mô hình có thể phát hiện các đối tượng xe hơi với độ chính xác cao trong các bối cảnh và điều kiện ánh sáng khác nhau.

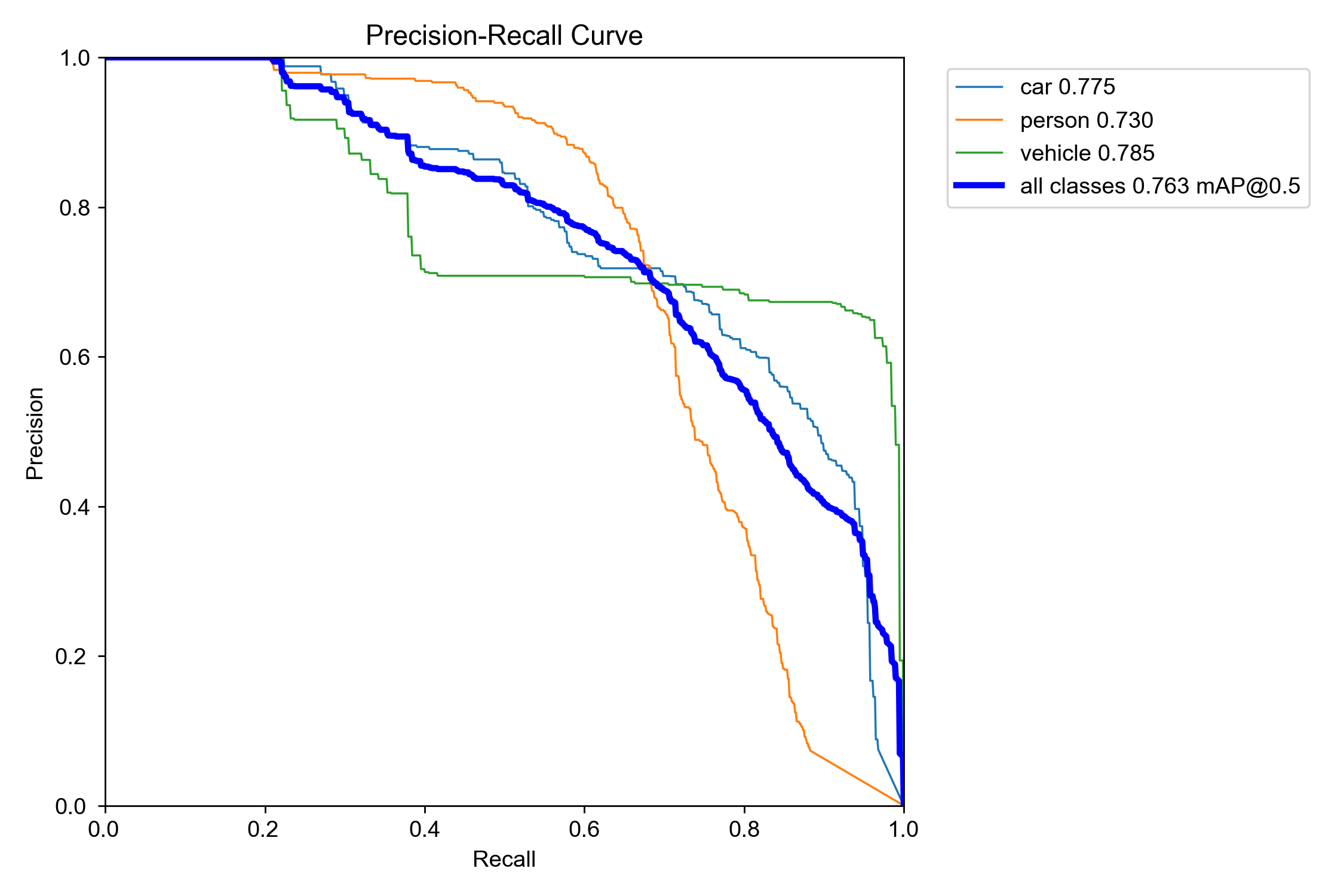
Person: 0.730 (73.0%)

Lớp "Person" có mAP50 là 73.0%, thấp hơn so với các lớp khác. Điều này cho thấy mặc dù mô hình vẫn có khả năng nhận diện con người, nhưng vẫn cần cải thiện thêm để tăng độ chính xác trong các tình huống như người bị che khuất, di chuyển nhanh, hoặc trong các cảnh nền phức tạp.

Vehicle: 0.785 (78.5%)

Mô hình đạt mAP50 cao nhất với lớp "Vehicle", lên tới 78.5%. Điều này phản ánh khả năng tốt của mô hình trong việc nhận diện các phương tiện giao thông, không chỉ ô tô mà còn có thể là các loại phương tiện khác như xe tải hoặc xe máy.

Mô hình đã đạt được kết quả tốt trong việc phát hiện các đối tượng phương tiện (car, vehicle) với mAP cao, đặc biệt là lớp "Vehicle" với 78.5%. Tuy nhiên, lớp "Person" cần cải thiện để nâng cao hiệu quả nhận diện con người trong các tình huống phức tạp, đặc biệt là khi có sự chồng lấp hoặc thay đổi góc nhìn. Tổng thể, mô hình đạt mAP50 là 76.3%, cho thấy khả năng dự đoán tốt nhưng vẫn có không gian cải thiện trong một số lớp đối tượng.



Hình 3.3: Đường cong Precision-Recall

#### *c. Đánh giá mAP50-95 (IoU từ 0.5 đến 0.95)*

mAP50-95 tổng thể: 0.576 (57.6%)

Mô hình đạt mAP50-95 57.6%, phản ánh hiệu suất giảm dần khi yêu cầu độ chồng lấp cao hơn giữa đối tượng dự đoán và thực tế.

Đánh giá theo từng lớp đối tượng:

Car: 0.553 (55.3%)

Mô hình nhận diện ô tô tốt ở mức IoU thấp, nhưng giảm hiệu suất khi yêu cầu độ chính xác cao hơn.

Person: 0.448 (44.8%)

Mô hình gặp khó khăn trong việc nhận diện con người khi yêu cầu độ chính xác cao.

Vehicle: 0.726 (72.6%)

Mô hình thể hiện hiệu suất tốt với lớp phương tiện, đặc biệt với độ chồng lấp cao.

#### *d. Đánh giá Accuracy (F1-score)*

Accuracy (F1-score) tổng thể: 0.747 (74.7%)

F1-score tổng thể đạt 74.7%, phản ánh sự cân bằng giữa Precision (73.2%) và Recall (76.3%).

Đánh giá theo từng lớp đối tượng:

Car: 0.706 (70.6%)

Mô hình có F1-score hợp lý trong việc nhận diện ô tô.

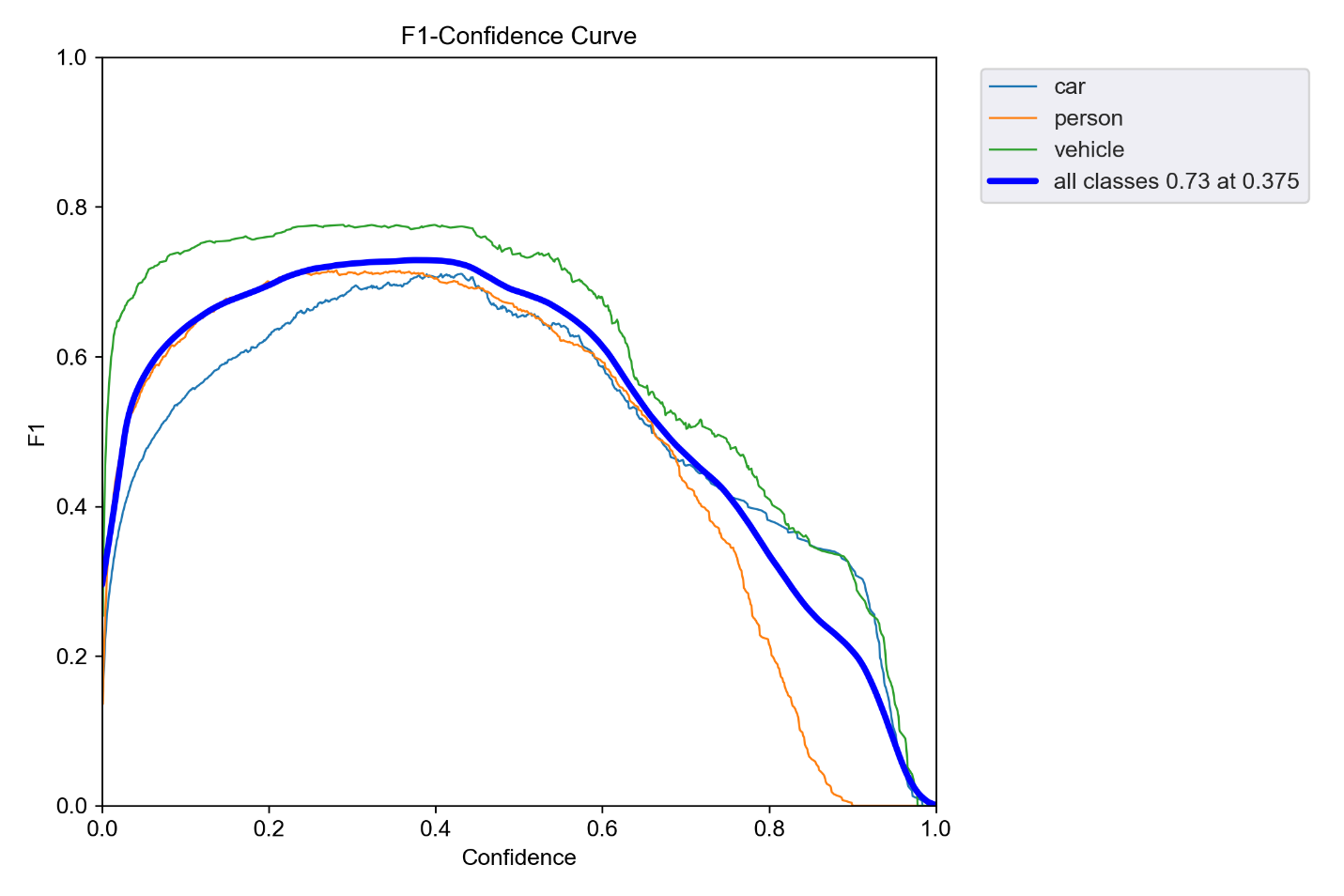
Person: 0.711 (71.1%)

F1-score cho thấy sự cân bằng tốt giữa độ chính xác và độ nhạy trong việc nhận diện con người.

Vehicle: 0.774 (77.4%)

F1-score cao cho thấy mô hình nhận diện phương tiện rất tốt.

Mô hình có F1-score và độ chính xác tổng thể tốt, đặc biệt là với lớp "Vehicle". Tuy nhiên, cần cải thiện hiệu suất ở các lớp "Car" và "Person" trong các tình huống khó khăn với yêu cầu độ chính xác cao hơn.

****

Hình 3.4: Đường cong Precision-Recall thể hiện hiệu suất qua các mức confidence.

## 3.3. Kết quả

Hệ thống được huấn luyện dựa trên các thông số kỹ thuật cụ thể, nhằm đảm bảo hiệu suất cao và khả năng học tốt của mô hình trong việc phát hiện và theo dõi đối tượng.

### 3.3.1. Thông số huấn luyện

#### a. Epochs:

Số lần huấn luyện toàn bộ tập dữ liệu: 50 epochs.

Mỗi epoch đại diện cho một lượt huấn luyện khi toàn bộ tập dữ liệu được đưa qua mô hình để tối ưu hóa các tham số. Việc huấn luyện với 50 epochs đảm bảo rằng mô hình có đủ thời gian để học và điều chỉnh các tham số nhằm đạt hiệu suất cao nhất, đồng thời tránh hiện tượng quá khớp (overfitting).

#### b. Batch size:

Kích thước lô dữ liệu: 16.

Batch size đề cập đến số lượng mẫu dữ liệu được xử lý cùng lúc trong một lần huấn luyện. Với batch size là 16, mô hình vừa đảm bảo sử dụng hiệu quả bộ nhớ GPU, vừa duy trì tốc độ huấn luyện ở mức tối ưu. Điều này phù hợp với kích thước của dữ liệu đầu vào và khả năng xử lý của phần cứng.

Kích thước ảnh đầu vào: 320x320 pixels.

Tất cả các ảnh trong tập dữ liệu đầu vào được chuẩn hóa về kích thước cố định 320x320 pixels. Kích thước này được lựa chọn dựa trên sự cân bằng giữa độ chính xác của mô hình và tốc độ xử lý.

#### c. Thời gian huấn luyện:

Tổng thời gian huấn luyện: 4.757 giờ.

Với tập dữ liệu lớn và số lượng epochs cao, thời gian huấn luyện kéo dài gần 5 giờ, cho thấy mức độ phức tạp của bài toán. Thời gian này được tối ưu nhờ sử dụng GPU mạnh mẽ để tăng tốc quá trình tính toán.

#### d. GPU memory sử dụng:

Dung lượng bộ nhớ GPU: ~3.06GB.

Đây là dung lượng bộ nhớ được sử dụng trong quá trình huấn luyện, bao gồm việc lưu trữ các tham số mô hình, dữ liệu đầu vào và kết quả trung gian trong các bước tính toán. Dung lượng này cho thấy mô hình sử dụng tài nguyên một cách hiệu quả để xử lý các bài toán phức tạp.

### 3.3.2 Kiến trúc mô hình YOLO

Kiến trúc của mô hình YOLO được thiết kế với độ sâu lớn và độ phức tạp cao, giúp mô hình có khả năng học các đặc trưng phức tạp và thực hiện các nhiệm vụ thời gian thực. Dưới đây là các thông tin chi tiết về kiến trúc:

#### a. Số lớp:

839 layers.

Kiến trúc của mô hình bao gồm 839 tầng nơ-ron (neural network layers). Đây là một mô hình rất sâu, cho phép hệ thống học và nhận diện được các đặc trưng tinh vi trong dữ liệu hình ảnh. Các tầng này bao gồm các lớp tích chập (convolutional layers), lớp gộp (pooling layers), và các lớp fully connected, giúp trích xuất thông tin từ ảnh và dự đoán bounding boxes cùng nhãn đối tượng.

#### b. Số tham số:

68,552,440 parameters.

Đây là tổng số trọng số (weights) và bias cần học trong suốt quá trình huấn luyện. Số lượng tham số lớn thể hiện khả năng học tập chi tiết của mô hình, giúp nhận diện chính xác các đối tượng trong điều kiện phức tạp. Tuy nhiên, nó cũng đòi hỏi tài nguyên tính toán mạnh mẽ để huấn luyện và triển khai.

#### c. Độ phức tạp tính toán

GFLOPs: 240.7 GFLOPs.

GFLOPs (Giga Floating Point Operations per Second) đo lường số phép tính dấu phẩy động cần thiết để chạy mô hình. Với giá trị 240.7 GFLOPs, mô hình có độ phức tạp tính toán cao, phù hợp với các bài toán yêu cầu hiệu suất mạnh như phát hiện đối tượng thời gian thực trong video.

Bộ thông số huấn luyện và kiến trúc của mô hình YOLO được thiết kế nhằm tối ưu hóa khả năng học và xử lý của hệ thống, đáp ứng yêu cầu khắt khe của các bài toán theo dõi đối tượng thời gian thực. Những thông số này đảm bảo rằng mô hình có thể hoạt động hiệu quả trên dữ liệu lớn và trong các điều kiện môi trường đa dạng.

### 3.3.3. Đánh giá kết quả:

#### a. Hiệu suất tổng thể

mAP50: 76.3%

mAP50-95: 57.6%

#### b. Hiệu suất theo từng lớp

Mô hình đạt hiệu suất tốt nhất với class Vehicle (Recall 92.6%, mAP50 78.5%)

Class Person có Precision cao (87.7%) nhưng Recall thấp (59.8%)

Class Car có hiệu suất ổn định (mAP50 77.5%)

****

Hình 3.5: Ma trận nhầm lẫn thể hiện hiệu suất phân loại của mô hình trên từng lớp dữ liệu.

### 3.3.4 Quá trình huấn luyện chi tiết

Quá trình huấn luyện của mô hình được giám sát chặt chẽ nhằm đảm bảo hiệu suất tối ưu, đồng thời tránh các vấn đề thường gặp như overfitting. Dưới đây là các điểm nổi bật về diễn biến của quá trình huấn luyện:

Loss giảm dần và ổn định sau 35 epochs

#### a. Diễn biến của loss function

Giai đoạn đầu: Loss giảm mạnh khi mô hình học nhanh từ dữ liệu.

Sau 35 epochs: Loss ổn định, cho thấy mô hình đã học được các đặc trưng quan trọng.

Ý nghĩa: Sự ổn định này đảm bảo mô hình không gặp khó khăn trong việc tối ưu hóa.

#### b. Tăng cường độ chính xác qua các epochs

Hiệu suất mô hình (mAP) tăng đều qua từng epoch.

Không có hiện tượng tụt giảm hoặc dao động mạnh, chứng tỏ các thông số huấn luyện đã được tối ưu tốt.

#### c. Phân tích hiện tượng Overfitting

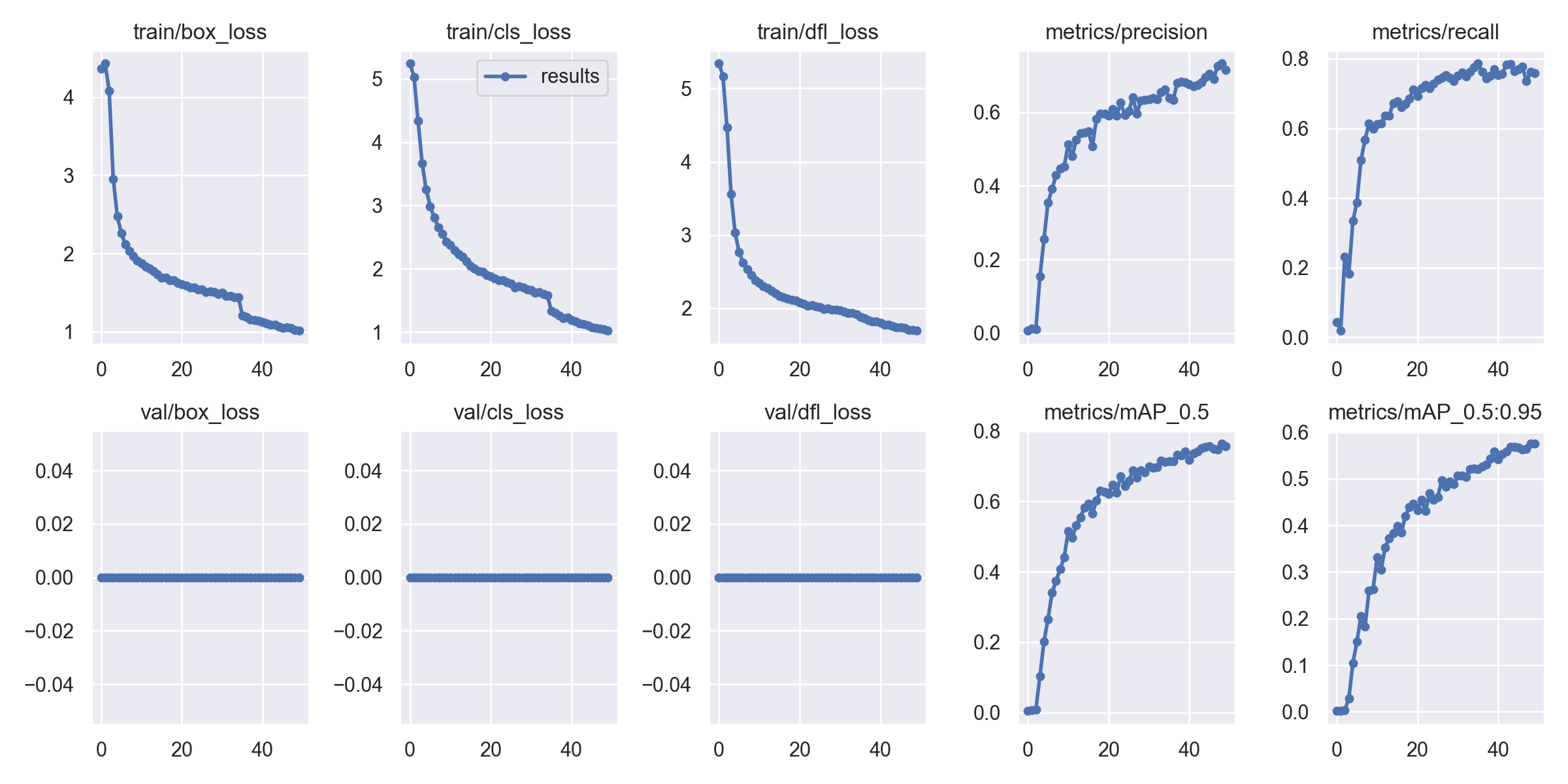
Không có dấu hiệu rõ rệt của overfitting:

Hiệu suất trên tập huấn luyện và kiểm thử tăng đều.

Đường cong loss giữa tập huấn luyện và validation gần như song song.

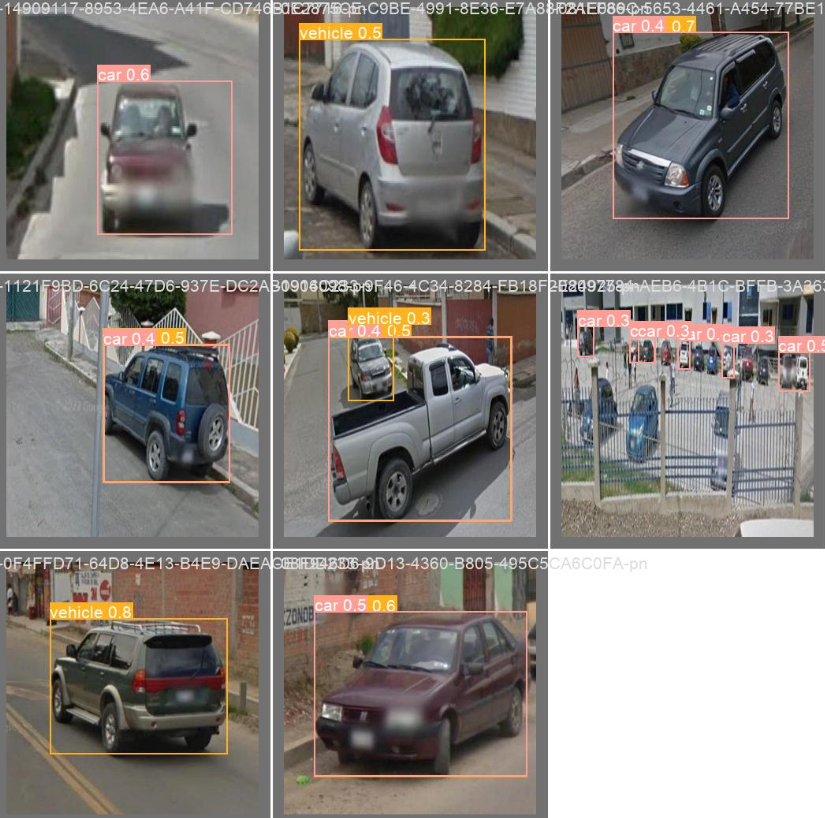
### 3.4.5 Kết luận về quá trình huấn luyện

Quá trình huấn luyện diễn ra thuận lợi với các chỉ số rõ ràng, bao gồm việc giảm loss, tăng hiệu suất đều đặn, và không xuất hiện overfitting. Điều này cho thấy mô hình đã được thiết lập và tối ưu hóa một cách hợp lý, đảm bảo khả năng áp dụng hiệu quả vào các ứng dụng thực tế mà không cần điều chỉnh quá nhiều.



Hình 3.6: Đồ thị loss

Hình ảnh nhận diện:

****

Hình 3.7: Các phương tiện được chương trình nhận diện thành công



Hình 3.8: Một số người còn mờ không nhận diện được

# KẾT LUẬN

**Kết quả đạt được**

Hệ thống theo dõi đối tượng trong video đã đạt được các kết quả quan trọng. Cụ thể, hệ thống hoạt động với độ chính xác cao, có khả năng nhận diện và theo dõi đối tượng hiệu quả trong nhiều điều kiện môi trường khác nhau như ánh sáng, góc quay và tốc độ di chuyển. Với tốc độ xử lý nhanh, hệ thống đảm bảo khả năng giám sát liên tục trong thời gian thực. Bên cạnh đó, nhờ sử dụng các công cụ như OpenCV và TensorFlow, hệ thống có cấu trúc gọn nhẹ, dễ dàng tích hợp với các nền tảng hiện có. Dữ liệu theo dõi được lưu trữ hiệu quả, hỗ trợ tốt cho việc phân tích hành vi và phát hiện các tình huống bất thường.

**Hướng phát triển**

Hệ thống cần được nâng cấp để có khả năng nhận diện và theo dõi nhiều đối tượng cùng lúc với độ chính xác cao hơn, đồng thời tích hợp các mô hình học sâu tiên tiến như YOLO hoặc TrackNet để tăng hiệu quả. Việc kết nối với các thiết bị IoT sẽ giúp gửi cảnh báo theo thời gian thực qua các kênh như Telegram hoặc Email. Ngoài ra, tính năng phân tích hành vi cần được bổ sung để phát hiện các tình huống bất thường như xâm nhập trái phép hoặc bỏ quên hành lý. Hệ thống cũng nên được tối ưu hóa để hoạt động hiệu quả trên các thiết bị phần cứng hạn chế như Raspberry Pi, đồng thời xây dựng giao diện trực quan, thân thiện để dễ dàng quản lý. Cuối cùng, hệ thống có thể mở rộng ứng dụng vào các lĩnh vực như giám sát giao thông, quản lý sản xuất, hoặc hỗ trợ chăm sóc người già và trẻ em, nhằm tăng tính thực tiễn và hiệu quả.

# TÀI LIỆU THAM KHẢO

[1] Lê Mỹ Hà (2019) Thị Giác máy tính và ứng dụng . Nhà xuất bản Đại Học Quốc Gia Hà Nội .Thành Phố Hồ Chí Minh

[2] Lê Thanh Hải. (2023). Xử lý ảnh và Thị giác máy tính. Hà Nội, Việt Nam: Nhà xuất bản Khoa học và Kỹ thuật.

[3] Lương Thị Hồng Lan (2024) Bài giảng Xử Lý Ảnh Và Thị Giác Máy Tính .Đại HỌc Công Nghệ Đông Á .Thành Phố Hà Nội

[4]Giáo trình Nhập môn Trí tuệ nhân tạo, Giáo sư Từ Minh Phương

[5] Bài Giảng Trường Đại Học Công Nghệ Đông Á

[6] Link tham khảo <https://universe.roboflow.com/>

[7] Link tham khảo <https://docs.ultralytics.com/vi/models/yolov9/>

[8] Link tham khảo https://github.com/WongKinYiu/yolov9