**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 14: XÂY DỰNG HỆ THỐNG XÁC NHẬN ĐỐI TƯỢNG VÀ ĐẾM ĐỐI TƯỢNG TRONG ẢNH**

**Giảng viên hướng dẫn: Lương Thị Hồng Lan**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Mã sinh viên** | **Sinh viên thực hiện** | **Lớp hành chính** |
| **1** | 20210591 | Cao Văn Hải | DCCNTT12.10.2 |
| **2** | 20210507 | Nguyễn Thị Phương Nam | DCCNTT12.10.2 |
| **3** | 20210452 | Trịnh Hữu Hoàng | DCCNTT12.10.2 |
| **4** | 20210599 | Nguyễn Thanh Hùng | DCCNTT12.10.2 |

**Bắc Ninh, năm 2024**

**Bắc Ninh, năm 2024**

**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 14: XÂY DỰNG HỆ THỐNG XÁC NHẬN ĐỐI TƯỢNG VÀ ĐẾM ĐỐI TƯỢNG TRONG ẢNH**

**Giảng viên hướng dẫn: Lương Thị Hồng Lan**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Mã sinh viên** | **Sinh viên thực hiện** | **Lớp hành chính** |
| **1** | 20210591 | Cao Văn Hải | DCCNTT12.10.2 |
| **2** | 20210507 | Nguyễn Thị Phương Nam | DCCNTT12.10.2 |
| **3** | 20210452 | Trịnh Hữu Hoàng | DCCNTT12.10.2 |
| **4** | 20210599 | Nguyễn Thanh Hùng | DCCNTT12.10.2 |

**Bắc Ninh, năm 2024**

**PHIẾU CHẤM THI BÀI TẬP LỚN**

**ĐỀ TÀI SỐ 14: XÂY DỰNG HỆ THỐNG XÁC NHẬN ĐỐI TƯỢNG VÀ ĐẾM ĐỐI TƯỢNG TRONG ẢNH**

| **TT** | **TIÊU CHÍ** | **THANG ĐIỂM** | **Cao Văn Hải** | **Nguyễn Thị Phương Nam** | **Trịnh Hữu Hoàng** | **Nguyễn Thanh Hùng** |
| --- | --- | --- | --- | --- | --- | --- |
| **20210591** | **20210507** | **20210452** | **20210599** |
| **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 |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Tiêu chí** | **Thang điểm** | **Cao Văn Hải** | **Nguyễn Thị Phương Nam** | **Trịnh Hữu Hoàng** | **Nguyễn Thanh Hùng** |
| **20210591** | **20210507** | **20210452** | **20210599** |
| **Tổng điểm bằng số:** | | 10 |  |  |  |  |
| **Tổng điểm bằng chữ:** | | Mười tròn |  |  |  |  |

**ĐIỂM BÀI TẬP LỚP**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Mã sinh viên** | **Sinh viên thực hiện** | **Điểm BTL** | |
| **Điểm số** | **Điểm chữ** |
| **1** | 20210591 | Cao Văn Hải |  |  |
| **2** | 20210507 | Phạm Thị Phương Nam |  |  |
| **3** | 20210452 | Trịnh Hữu Hoàng |  |  |
| **4** | 20210559 | Nguyễn Thanh Hùng |  |  |

|  |  |
| --- | --- |
| **Cán bộ chấm thi 1**  *(Ký và ghi rõ họ tên)* | **Cán bộ chấm thi 2**  *(Ký và ghi rõ họ tên)* |

**MỤC LỤC**

[**LỜI MỞ ĐẦU** 8](#_Toc184589176)

[**LỜI CẢM ƠN** 10](#_Toc184589177)

[**CHƯƠNG I. CÁC KIẾN THỨC CƠ SỞ** 11](#_Toc184589178)

[**1.1 Nhận dạng đối tượng** 11](#_Toc184589179)

[**1.1.1 Mô tả bài toán** 11](#_Toc184589180)

[**1.1.2 Khó khăn trong bài toán** 13](#_Toc184589181)

[**1.1.3.Ứng dụng** 14](#_Toc184589182)

[**1.2 Các kỹ thuật sử dụng trong bài toán nhận dạng** 19](#_Toc184589183)

[**1.2.1 Support Vector Machine (SVM)** 19](#_Toc184589184)

[**1.2.2 K - Nearest Neighbors (K - NN)** 20](#_Toc184589185)

[**1.2.3 Decision Tree** 20](#_Toc184589186)

[**1.3 Ngôn ngữ lập trình và các thư viện sử dụng** 21](#_Toc184589187)

[**1.3.1 Python** 21](#_Toc184589188)

[**1.3.2 Visual Studio Code** 23](#_Toc184589189)

[**1.3.3 Các thư viện sử dụng** 25](#_Toc184589190)

[**CHƯƠNG II. XÂY DỰNG HỆ THỐNG** 26](#_Toc184589191)

[**2.1 Yêu cầu chức năng** 26](#_Toc184589192)

[**2.1.1 Mô tả bài toán** 26](#_Toc184589193)

[**2.1.2 Yêu cầu của bài toán** 27](#_Toc184589194)

[**2.1.3 Input (Đầu vào)** 27](#_Toc184589195)

[**2.1.4 Output (Đầu ra)** 28](#_Toc184589196)

[**2.2 Xây dựng hệ thống** 29](#_Toc184589197)

[**2.2.1 Mô tả các bước trong sơ đồ:** 29](#_Toc184589198)

[**2.2.2 Sơ đồ đếm đối tượng** 31](#_Toc184589199)

[**CHƯƠNG III. KẾT QUẢ THỰC NGHIỆM** 32](#_Toc184589200)

[**3.1 Dữ liệu** 32](#_Toc184589201)

[**3.1.1 Nguồn Dữ liệu** 32](#_Toc184589202)

[**3.1.2** **Tiền xử lý Dữ liệu** 32](#_Toc184589203)

[**3.2 Các độ đo** 33](#_Toc184589204)

[**3.2.1. Độ Chính Xác (Accuracy)** 33](#_Toc184589205)

[**3.2.2.** **Precision và Recall** 35](#_Toc184589206)

[**3.2.3. F1 Score** 37](#_Toc184589207)

[**3.2.4. Mean Average Precision (mAP)** 38](#_Toc184589208)

[**3.3 Kết quả thực nghiệm** 40](#_Toc184589209)

[**KẾT LUẬN** 41](#_Toc184589210)

[**TÀI LIỆU THAM KHẢO** 43](#_Toc184589211)

# **LỜI MỞ ĐẦU**

Trong thời đại cách mạng công nghiệp 4.0, sự phát triển mạnh mẽ của công nghệ đã tạo ra những bước tiến vượt bậc trong nhiều lĩnh vực, đặc biệt là trí tuệ nhân tạo (AI) và thị giác máy tính (Computer Vision). Các ứng dụng dựa trên thị giác máy tính không chỉ dừng lại ở việc phân tích hình ảnh mà còn mang đến các giải pháp vượt trội cho các vấn đề trong đời sống, sản xuất và nghiên cứu khoa học. Một trong những ứng dụng tiêu biểu của lĩnh vực này là việc xác nhận và đếm số lượng đối tượng trong ảnh – một bài toán có tính ứng dụng cao trong nhiều ngành nghề như an ninh, giao thông, quản lý kho bãi, nông nghiệp thông minh, và thương mại điện tử.

Đề tài "Xây dựng hệ thống xác nhận đối tượng và đếm đối tượng trong ảnh" ra đời nhằm giải quyết các thách thức đặt ra trong bài toán này. Hệ thống được thiết kế để tự động nhận diện các loại đối tượng có trong hình ảnh, xác định chính xác vị trí và số lượng của chúng thông qua việc ứng dụng các thuật toán học máy (Machine Learning) và học sâu (Deep Learning). Điều này không chỉ giúp giảm thiểu thời gian và công sức mà con người phải bỏ ra trong việc phân tích hình ảnh, mà còn cải thiện đáng kể độ chính xác và hiệu quả của các hệ thống hiện tại.

Mục tiêu của đề tài không chỉ dừng lại ở việc xây dựng một hệ thống hoạt động ổn định, mà còn hướng đến khả năng mở rộng để đáp ứng các yêu cầu thực tế đa dạng. Hệ thống được kỳ vọng sẽ đạt được sự cân bằng giữa độ chính xác, thời gian xử lý và tính ứng dụng trong thực tiễn. Các thuật toán và mô hình sẽ được lựa chọn và tối ưu hóa kỹ lưỡng, dựa trên dữ liệu thực tế và các tiêu chí đánh giá khoa học như độ chính xác, độ nhạy, độ đặc hiệu, và khả năng tổng quát hóa.

Bên cạnh đó, đề tài cũng đặt mục tiêu khám phá và phát triển các kỹ thuật tiên tiến, tận dụng tối đa sức mạnh của công nghệ xử lý hình ảnh và học sâu. Điều này sẽ không chỉ mang lại giá trị cho việc thực hiện đề tài mà còn mở ra nhiều tiềm năng nghiên cứu và ứng dụng trong tương lai.

Với tính cấp thiết và tính ứng dụng cao của bài toán, chúng tôi tin tưởng rằng đề tài sẽ mang lại những đóng góp tích cực cho lĩnh vực xử lý ảnh và trở thành nền tảng cho những hệ thống tự động hóa thông minh hơn trong tương lai.

# **LỜI CẢM ƠN**

Đầu tiên, chúng em xin gửi lời cảm ơn chân thành đến Trường Đại học Công nghệ Đông Á. Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến giảng viên bộ môn Cô Lương Thị Hồng Lan đã dạy dỗ, truyền đạt những kiến thức quý báu cho chúng em trong suốt thời gian học tập vừa qua. Trong thời gian tham gia lớp học của cô, chúng em đã có thêm cho mình nhiều kiến thức bổ ích, tinh thần học tập hiệu quả, nghiêm túc. Đây chắc chắn sẽ là những kiến thức quý báu, là hành trang để em có thể vững bước sau này.

Bộ môn Xử lý ảnh và thị giác máy tính là môn học thú vị, vô cùng bổ ích và có tính thực tế cao. Đảm bảo cung cấp đủ kiến thức, gắn liền với nhu cầu thực tiễn của sinh viên. Tuy nhiên, do vốn kiến thức còn nhiều hạn chế và khả năng tiếp thu thực tế còn nhiều bỡ ngỡ. Mặc dù chúng em đã cố gắng hết sức nhưng chắc chắn bài tiểu luận khó có thể tránh khỏi những thiếu sót và nhiều chỗ còn chưa chính xác, kính mong cô xem xét và góp ý để bài tiểu luận của em được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

# **CHƯƠNG I. CÁC KIẾN THỨC CƠ SỞ**

## **1.1 Nhận dạng đối tượng**

Hệ thống xác nhận đối tượng và đếm đối tượng trong ảnh là một bài toán quan trọng trong lĩnh vực thị giác máy tính, với mục tiêu phát hiện, nhận diện, và theo dõi sự di chuyển của một hoặc nhiều đối tượng qua nhiều khung hình liên tiếp. Bài toán này yêu cầu đảm bảo sự nhất quán trong việc theo dõi đối tượng, ngay cả khi có sự thay đổi về góc quay, kích thước, hoặc sự xuất hiện của các yếu tố gây nhiễu trong video.

### **1.1.1 Mô tả bài toán**

Bài toán đặt ra là xây dựng một hệ thống phát hiện và nhận diện đối tượng trong video thời gian thực, với các yêu cầu cụ thể:

**Mục tiêu bài toán:**

Xác nhận đối tượng: Phát hiện tất cả các đối tượng trong ảnh, phân loại chúng vào các danh mục đã định trước (người, xe hơi, chó, mèo,...).

Đếm đối tượng: Đếm số lượng đối tượng theo từng loại đã phát hiện được.

**Phát hiện nhanh và chính xác**:

           Hệ thống cần xử lý liên tục các khung hình từ ảnh, đảm bảo phát hiện đối tượng ngay lập tức.

           Tốc độ xử lý cần đạt thời gian thực.

**Xử lý trong các bối cảnh phức tạp**:

           Nhận diện chính xác ngay cả khi đối tượng bị che khuất một phần, xuất hiện trong môi trường ánh sáng yếu, bị lóa hoặc bị nhiễu.

**Yêu cầu dữ liệu đầu vào**

Các bức ảnh có chứa các đối tượng cần phát hiện.

Dữ liệu nên phong phú về ánh sáng, góc nhìn, kích thước và sự che khuất đối tượng.

**Dữ liệu gán nhãn**

Mỗi bức ảnh cần được gắn nhãn chi tiết, bao gồm:

Loại đối tượng: Ví dụ, "xe hơi", "người".

Bounding Box hoặc Polygon: Xác định vùng đối tượng trong ảnh.

Công cụ hỗ trợ gán nhãn:

LabelImg (miễn phí, phổ biến).

CVAT (Công cụ trực tuyến mạnh mẽ).

**Đầu ra trực quan**:

           Hiển thị bounding boxes với nhãn đối tượng trên khung hình video.

           Cung cấp thông tin chi tiết về độ tin cậy và số lượng đối tượng được nhận diện.

**Huấn luyện mô hình:**

**Tách dữ liệu**: 80% huấn luyện, 10% kiểm tra, 10% đánh giá.

**Cài đặt môi trường**:

Sử dụng Python, TensorFlow hoặc PyTorch.

GPU: CUDA cho tăng tốc huấn luyện.

**Thông số huấn luyện**:

Epoch: Số lần chạy qua toàn bộ tập dữ liệu.

Batch Size: Số lượng ảnh xử lý cùng lúc.

Learning Rate: Điều chỉnh tốc độ học.

**Đánh giá**:

Sử dụng **mAP** (Mean Average Precision) để kiểm tra độ chính xác.

Tối ưu hóa mô hình bằng cách thay đổi tham số.

### **1.1.2 Khó khăn trong bài toán**

Nhận dạng đối tượng đối tượng và đếm đối tượng trong ảnh đối mặt với nhiều thách thức phức tạp, bao gồm:

**Biến đổi về tư thế và góc nhìn:** Đối tượng có thể xuất hiện ở nhiều tư thế và góc nhìn khác nhau, dẫn đến sự thay đổi đáng kể trong hình ảnh thu được. Ví dụ, khuôn mặt người có thể được chụp từ phía trước, bên cạnh hoặc từ trên xuống, gây khó khăn cho việc nhận dạng chính xác.

**Sự che khuất và thiếu hụt thông tin:** Đối tượng có thể bị che khuất một phần hoặc hoàn toàn bởi các vật thể khác, làm giảm lượng thông tin cần thiết cho việc nhận dạng. Chẳng hạn, một chiếc xe có thể bị che khuất bởi cây cối hoặc các phương tiện khác trong ảnh.

**Biến dạng và thay đổi hình dạng:** Đối tượng có thể thay đổi hình dạng do các yếu tố như chuyển động, biến dạng vật lý hoặc do góc nhìn. Ví dụ, một chiếc ô tô có thể trông khác nhau khi nhìn từ phía trước so với từ phía sau.

**Sự phức tạp của nền và nhiễu:** Hình nền phức tạp hoặc nhiễu trong ảnh có thể gây nhầm lẫn cho hệ thống nhận dạng, đặc biệt khi đối tượng và nền có màu sắc hoặc kết cấu tương tự nhau. Ví dụ, nhận dạng một con mèo màu xám trên nền bê tông xám có thể gặp khó khăn.

**Điều kiện ánh sáng và chất lượng hình ảnh:** Sự thay đổi về ánh sáng, bóng đổ hoặc chất lượng hình ảnh kém (như độ phân giải thấp, mờ) có thể ảnh hưởng tiêu cực đến khả năng nhận dạng. Ví dụ, nhận dạng khuôn mặt trong điều kiện ánh sáng yếu hoặc ngược sáng thường gặp nhiều khó khăn.

**Đa dạng về chủng loại và biến thể của đối tượng:**

Đối tượng có thể có nhiều biến thể về màu sắc, kích thước, hình dạng hoặc kiểu dáng, đòi hỏi hệ thống phải có khả năng nhận dạng chính xác trong mọi trường hợp. Ví dụ, nhận dạng các loại xe ô tô với nhiều mẫu mã và màu sắc khác nhau.

**Thời gian xử lý và yêu cầu tính toán:**

Đối với ứng dụng thời gian thực, hệ thống phải xử lý nhanh chóng và hiệu quả, đòi hỏi tài nguyên tính toán mạnh mẽ và thuật toán tối ưu. Ví dụ, trong hệ thống giám sát an ninh, việc nhận dạng đối tượng phải diễn ra tức thì để kịp thời phản ứng.

### **1.1.3.Ứng dụng**

**Giao thông và vận tải**

Phát hiện và đếm phương tiện giao thông:

Mục đích: Đánh giá mật độ giao thông, dự báo tắc đường, tối ưu hóa tín hiệu giao thông.

Ví dụ: Một camera giao thông đếm số lượng xe hơi, xe máy, xe buýt đi qua trong một khoảng thời gian.

Lợi ích:

Cải thiện quản lý giao thông đô thị.

Phát hiện các hành vi vi phạm (đi vào làn xe cấm, dừng đỗ sai quy định).

Hệ thống thu phí tự động (ETC):

Mục đích: Xác định loại xe để áp dụng phí đường bộ phù hợp.

**Bán lẻ và thương mại**

Phân tích hành vi khách hàng:

Mục đích: Đếm số lượng người trong cửa hàng để phân tích lưu lượng khách.

Ví dụ: Một siêu thị đếm số người tại các khu vực khác nhau để tối ưu bố trí hàng hóa và nhân viên.

Kiểm kê tự động trong kho hàng:

Mục đích: Đếm số lượng sản phẩm, hàng hóa trên kệ mà không cần kiểm tra thủ công.

Ví dụ: Sử dụng camera trên robot tự động hoặc drone.

Lợi ích: Giảm sai sót và tăng tốc quá trình kiểm kê.

**Sản xuất công nghiệp**

Giám sát dây chuyền sản xuất:

Mục đích: Phát hiện và đếm số lượng sản phẩm trên dây chuyền để đảm bảo sản lượng.

Ví dụ: Một nhà máy sản xuất wifi đếm số lượng thiết bị đi qua từng giai đoạn lắp ráp.

Phát hiện lỗi sản phẩm:

Mục đích: Xác định sản phẩm bị lỗi (bị méo, vỡ, thiếu linh kiện).

**Nông nghiệp thông minh**

Đếm vật nuôi:

Mục đích: Theo dõi số lượng động vật trong trang trại.

Ví dụ: Camera phát hiện và đếm bò hoặc gà trong một chuồng lớn.

Lợi ích: Theo dõi sức khỏe và quản lý số lượng chính xác.

Đếm cây trồng hoặc trái cây:

Mục đích: Giám sát năng suất vườn cây, tính toán lượng phân bón hoặc nước tưới.

Ví dụ: Drone chụp ảnh vườn cây để đếm số lượng quả cam hoặc táo.

**An ninh và giám sát**

Phát hiện người xâm nhập:

Mục đích: Phát hiện người xâm nhập vào các khu vực bị cấm.

Ví dụ: Camera tại công trình hoặc kho phát hiện và cảnh báo khi có người lạ xuất hiện.

Đếm số lượng người trong khu vực:

Mục đích: Theo dõi số lượng người trong các tòa nhà hoặc sự kiện đông đúc để đảm bảo an toàn.

Ví dụ: Giám sát đám đông trong sân vận động hoặc nhà ga.

**Y tế và chăm sóc sức khỏe**

Giám sát bệnh nhân:

Mục đích: Theo dõi bệnh nhân trong các khu vực cách ly hoặc chăm sóc đặc biệt.

Ví dụ: Hệ thống camera phát hiện và đếm bệnh nhân trong một phòng cách ly.

Quản lý cơ sở y tế:

Mục đích: Đếm số lượng người trong phòng khám, bệnh viện để tối ưu hóa dịch vụ chăm sóc.

**Môi trường và bảo tồn thiên nhiên**

Theo dõi động vật hoang dã:

Mục đích: Đếm và giám sát số lượng động vật trong môi trường tự nhiên.

Ví dụ: Dùng camera tại các khu bảo tồn để đếm số lượng voi hoặc hổ.

Quản lý rác thải:

Mục đích: Phát hiện và đếm các loại rác thải tại bãi rác hoặc khu vực công cộng.

**Giáo dục và quản lý trường học**

Điểm danh tự động:

Mục đích: Đếm số lượng học sinh trong lớp học.

Lợi ích: Tiết kiệm thời gian và đảm bảo tính chính xác.

Quản lý an ninh:

Mục đích: Phát hiện người lạ ra vào trường học.

**Giải trí và sự kiện**

Phân tích đám đông:

Mục đích: Theo dõi số lượng người tham gia sự kiện, như lễ hội, buổi hòa nhạc.

Lợi ích: Đảm bảo an toàn, tránh quá tải tại một khu vực.

Đếm khách tại công viên giải trí:

Mục đích: Đếm số lượng khách để quản lý dịch vụ và tối ưu hóa trải nghiệm.

**Ứng dụng quân sự**

Giám sát biên giới:

Mục đích: Phát hiện và đếm các đối tượng (người, phương tiện) vượt qua biên giới.

Quản lý thiết bị quân sự:

Mục đích: Đếm và theo dõi các loại vũ khí, phương tiện quân sự.

## **1.2 Các kỹ thuật sử dụng trong bài toán nhận dạng**

### **1.2.1 Support Vector Machine (SVM)**

**Tư tưởng kỹ thuật:**

SVM là một kỹ thuật học máy được sử dụng để phân loại dữ liệu bằng cách tìm một siêu phẳng tối ưu phân tách hai lớp dữ liệu sao cho khoảng cách từ các điểm gần nhất của mỗi lớp đến siêu phẳng này (margin) là lớn nhất.  
Nếu dữ liệu không tuyến tính, SVM sử dụng các kernel (nhân) để ánh xạ dữ liệu vào không gian chiều cao hơn, nơi chúng có thể phân tách được.

**Hình mô tả kỹ thuật:**

Một hình minh họa gồm hai lớp dữ liệu (chấm xanh và đỏ) và siêu phẳng tối ưu cùng với margin.  
*(Mô tả nếu bạn cần hình: hai nhóm điểm màu xanh và đỏ phân biệt bằng một đường thẳng ở giữa, hai đường nét đứt song song biểu diễn khoảng margin.)*

**Ưu điểm:**

Hiệu quả với dữ liệu có số chiều cao.

Sử dụng kernel linh hoạt cho bài toán không tuyến tính.

Có khả năng tổng quát tốt trên tập dữ liệu nhỏ.

**Nhược điểm:**

Hiệu quả giảm với tập dữ liệu lớn, đa lớp.

Yêu cầu điều chỉnh kernel phù hợp.

### **1.2.2 K - Nearest Neighbors (K - NN)**

**Tư tưởng kỹ thuật:**

KNN là thuật toán đơn giản, dựa trên khoảng cách. Khi cần phân loại một điểm mới, thuật toán tìm k điểm gần nhất trong dữ liệu huấn luyện và dựa vào nhãn của chúng (đa số phiếu) để quyết định nhãn của điểm đó.

**Hình mô tả kỹ thuật:**

Một tập hợp các điểm thuộc hai lớp màu xanh và đỏ. Một điểm mới nằm giữa các điểm này, được phân loại dựa vào khoảng cách đến 3 điểm gần nhất.  
(Mô tả: điểm tròn xanh dương với đường nét đứt chỉ đến các điểm gần nhất thuộc các nhóm khác nhau.)

**Ưu điểm:**

Dễ triển khai, không cần giai đoạn huấn luyện phức tạp.

Hiệu quả trong các tập dữ liệu nhỏ, đơn giản.

**Nhược điểm:**

Hiệu quả giảm khi dữ liệu lớn và phân bố phức tạp.

Yêu cầu xác định k thích hợp.

### **1.2.3 Decision Tree**

**Tư tưởng kỹ thuật:**

Decision Tree sử dụng cấu trúc cây để đưa ra quyết định. Tại mỗi nút, thuật toán sẽ chọn thuộc tính tốt nhất để phân chia dữ liệu sao cho thông tin hoặc độ lợi lớn nhất, tạo nhánh dẫn đến kết quả cuối cùng.

**Hình mô tả kỹ thuật:**

Một cây phân loại với các nút chứa điều kiện (ví dụ: "Chiều dài > 5 cm?") và nhánh dẫn đến các lớp đầu ra.  
*(Mô tả: cây nhị phân với nhãn "Có" hoặc "Không" cho từng điều kiện và dẫn đến kết quả cuối.)*

**Ưu điểm:**

Dễ hiểu, dễ giải thích.

Hỗ trợ cả dữ liệu phân loại và liên tục.

Tự động xử lý các thuộc tính không quan trọng.

**Nhược điểm:**

Dễ bị quá khớp nếu cây quá phức tạp.

Không ổn định khi có thay đổi nhỏ trong dữ liệu.

## **1.3 Ngôn ngữ lập trình và các thư viện sử dụng**

### **1.3.1 Python**

**Python**: Là ngôn ngữ phổ biến cho xử lý ảnh và thị giác máy tính nhờ vào cú pháp dễ đọc và hệ sinh thái thư viện phong phú.

**Tổng quan về python**: Python được thiết kế như một ngôn ngữ lập trình đa dụng (general-purpose), có thể áp dụng trong nhiều lĩnh vực như. Lập trình web (Web Development) và Xử lý ảnh/video (Image/Video Processing)

**Tính năng nổi bật của Python:**

**Dễ học và dễ sử dụng:**

Cú pháp gần gũi, ngắn gọn, dễ đọc.

Giảm thiểu các ký hiệu không cần thiết (ví dụ: không cần dấu chấm phẩy ;).

**Hướng đối tượng và đa mô hình:**

Hỗ trợ lập trình hướng đối tượng (OOP), hướng thủ tục, và hàm (Functional Programming).

**Thông dịch:**

Python là ngôn ngữ thông dịch (interpreted), không cần biên dịch (compilation), giúp giảm thời gian chạy thử.

**Thư viện phong phú:**

Python cung cấp hàng nghìn thư viện chuẩn và bên thứ ba để giải quyết đa dạng các bài toán từ nhỏ đến lớn.

**Mã nguồn mở:**

Python hoàn toàn miễn phí và mã nguồn mở, cho phép cộng đồng phát triển liên tục.

**Tương thích đa nền tảng:**

Chạy được trên các hệ điều hành phổ biến như Windows, macOS, Linux.

**Hỗ trợ tích hợp tốt:**

Python có thể tích hợp dễ dàng với các ngôn ngữ khác như C, C++, Java.

**Ưu điểm của Python:**

**Ngôn ngữ đa năng:**

Python có thể sử dụng trong nhiều lĩnh vực khác nhau, từ khoa học dữ liệu đến lập trình web.

**Cộng đồng hỗ trợ mạnh:**

Với cộng đồng lớn, người dùng dễ dàng tìm được tài liệu, hướng dẫn, và giải pháp cho các vấn đề lập trình.

**Khả năng mở rộng:**

Dễ dàng tích hợp với các thư viện và công cụ khác (ví dụ: viết các module hiệu năng cao bằng C/C++).

**Hiệu suất cao cho prototyping:**

Python giúp phát triển nhanh chóng các nguyên mẫu (prototypes) nhờ cú pháp ngắn gọn và thư viện phong phú.

**Hỗ trợ khoa học và học thuật:**

Python được sử dụng phổ biến trong các lĩnh vực nghiên cứu, học thuật với các thư viện như NumPy, SciPy, Matplotlib.

**Tính linh hoạt cao:**

Có thể dùng Python để phát triển cả ứng dụng nhỏ lẫn hệ thống lớn.

### **1.3.2 Visual Studio Code**

Trong đề tài nhận diện và đếm đối tượng trong ảnh, Visual Studio được sử dụng làm môi trường phát triển tích hợp (IDE - Integrated Development Environment) chính để lập trình và triển khai các giải pháp liên quan đến xử lý ảnh và học máy. Visual Studio, phát triển bởi Microsoft, là một trong những IDE mạnh mẽ nhất hỗ trợ ngôn ngữ Python, đặc biệt phù hợp cho các dự án liên quan đến xử lý ảnh và trí tuệ nhân tạo.

**Tính năng nổi bật:**

Soạn thảo mã thông minh: Hỗ trợ tự động hoàn thành mã, kiểm tra lỗi cú pháp, và tái cấu trúc mã nhanh chóng.

Quản lý dự án: Cung cấp giao diện trực quan để tổ chức tệp tin và thư mục, hỗ trợ tích hợp với các thư viện.

Vai trò của Visual Studio trong đề tài Trong bài toán nhận diện và đếm đối tượng, Visual Studio hỗ trợ các khía cạnh quan trọng sau:

Tạo môi trường lập trình với Python: Visual Studio cung cấp các công cụ giúp dễ dàng quản lý dự án Python, thiết lập cấu hình, và thêm các thư viện cần thiết.

Quản lý cấu trúc dự án: Giao diện trực quan giúp quản lý các lớp, tệp, và module trong dự án, đảm bảo cấu trúc rõ ràng và dễ mở rộng.

Tích hợp Git hỗ trợ kiểm soát phiên bản và làm việc nhóm hiệu quả.

Triển khai và tối ưu hóa: Visual Studio hỗ trợ biên dịch mã Python thành các ứng dụng độc lập hoặc triển khai trên nền tảng đám mây như Azure.

**Ưu điểm của Visual Studio trong đề tài**

Hiệu suất cao: Được tối ưu hóa để phát triển các ứng dụng .NET, giúp tăng tốc độ phát triển và triển khai.

Hỗ trợ mạnh mẽ: Kết nối liền mạch với các công cụ như Azure và Docker

### **1.3.3 Các thư viện sử dụng**

**OpenCV**

Mô tả: Thư viện mã nguồn mở mạnh mẽ cho xử lý ảnh và video.

Chức năng: Phát hiện đối tượng, xử lý hình ảnh, phân đoạn ảnh.

**TensorFlow/Keras**

Mô tả: Thư viện cho học sâu, hỗ trợ xây dựng và huấn luyện mô hình nhận diện.

Chức năng: Phát hiện và phân loại đối tượng.

**PyTorch**

Mô tả: Thư viện học sâu khác rất phổ biến, thích hợp cho nghiên cứu.

Chức năng: Xây dựng mô hình nhận diện đối tượng.

**Scikit-image**

Mô tả: Thư viện cho xử lý ảnh trong Python.

Chức năng: Cung cấp nhiều công cụ cho phân tích và xử lý hình ảnh.

**Detectron2**

Mô tả: Thư viện từ Facebook AI Research dành cho phát hiện và phân đoạn đối tượng.

Chức năng: Hỗ trợ nhiều mô hình hiện đại cho xác nhận và đếm đối tượng.

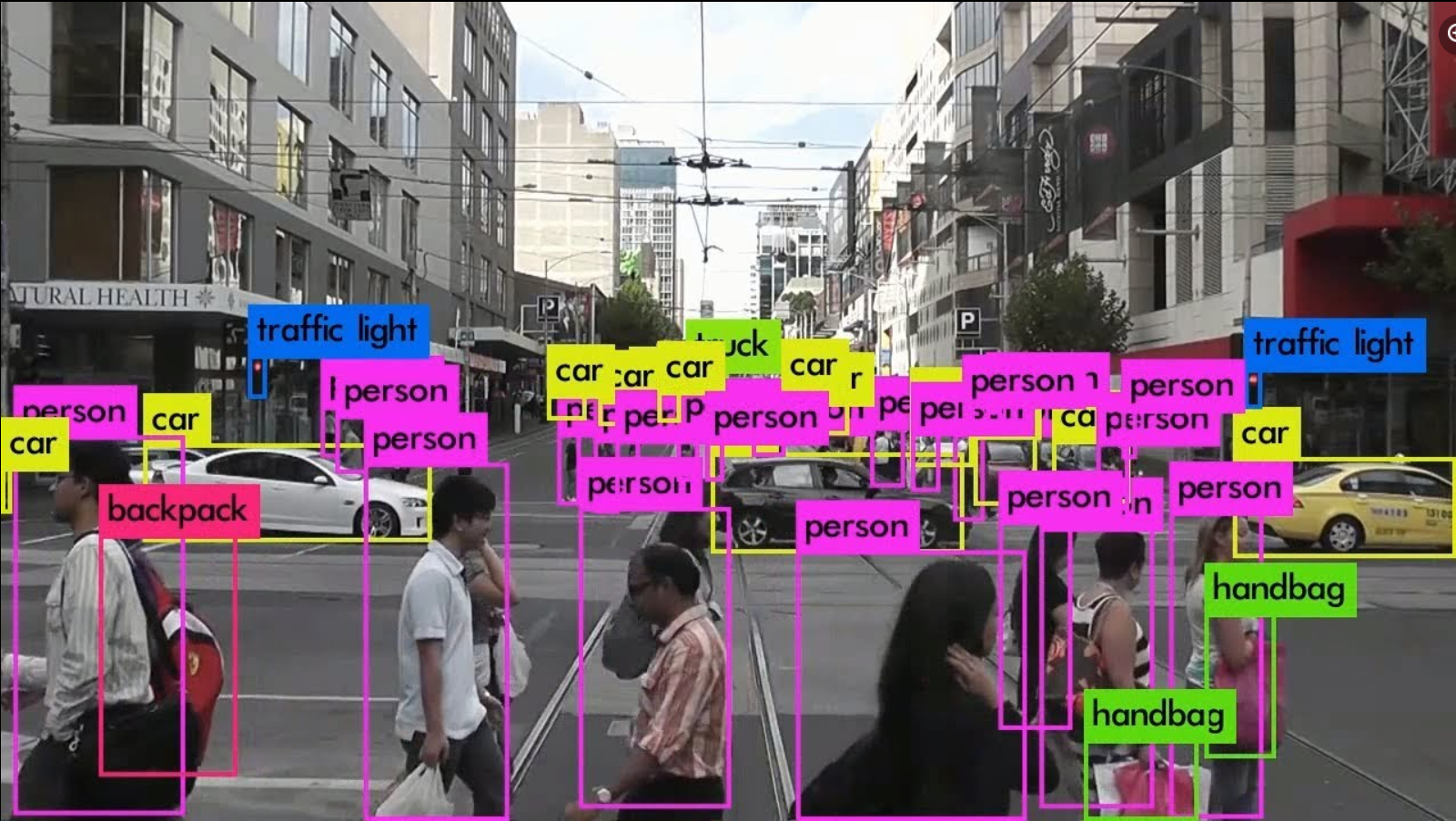
**YOLO (You Only Look Once)**

Mô tả: Một trong những mô hình phát hiện đối tượng nhanh nhất.

Chức năng: Phát hiện và đếm đối tượng trong thời gian thực.

# **CHƯƠNG II. XÂY DỰNG HỆ THỐNG**

## **2.1 Yêu cầu chức năng**



*Hình 2.1 Hình ảnh minh họa*

### **2.1.1 Mô tả bài toán**

Bài toán yêu cầu xây dựng một hệ thống nhận dạng và phân loại đối tượng trong ảnh. Với một hình ảnh bất kỳ, hệ thống cần:

**Xác định các đối tượng xuất hiện trong ảnh** dựa trên các danh mục được định nghĩa trước (ví dụ: người, xe hơi, đèn giao thông, túi xách, v.v.).

**Khoanh vùng đối tượng** bằng cách vẽ các khung bao (bounding boxes) xung quanh chúng.

**Gán nhãn cho từng đối tượng** theo đúng danh mục.

**Đếm số lượng đối tượng** của từng loại trong ảnh.

### **2.1.2 Yêu cầu của bài toán**

**Độ chính xác cao:**

Hệ thống phải phân loại đúng các đối tượng trong ảnh với tỷ lệ chính xác ≥ 90%.

**Phát hiện đa đối tượng:**

Hỗ trợ nhận dạng nhiều đối tượng trong cùng một ảnh, kể cả khi các đối tượng chồng lấp nhau.

**Thời gian xử lý hợp lý:**

Đảm bảo hệ thống hoạt động nhanh, với thời gian xử lý dưới 5 giây cho mỗi ảnh (với ảnh độ phân giải trung bình).

**Hỗ trợ đa danh mục:**

Hệ thống phải nhận dạng được nhiều loại đối tượng khác nhau (ví dụ: người, xe cộ, túi xách, đèn giao thông).

### **2.1.3 Input (Đầu vào)**

Một bức ảnh bất kỳ, có thể chứa một hoặc nhiều đối tượng (ví dụ: ảnh đường phố, ảnh trong nhà, v.v.).

Định dạng ảnh: JPEG, PNG, BMP.

Ví dụ từ hình ảnh cung cấp:

Đầu vào là một bức ảnh đường phố, chứa nhiều đối tượng như người, ô tô, túi xách, và đèn giao thông.

### **2.1.4 Output (Đầu ra)**

**Hình ảnh đã xử lý:**

Các đối tượng trong ảnh được khoanh vùng bởi khung bao (bounding boxes).

Nhãn phân loại được hiển thị trên mỗi đối tượng (ví dụ: "person", "car", "traffic light").

**Bảng thống kê (tuỳ chọn):**

Số lượng từng loại đối tượng phát hiện được trong ảnh.

Ví dụ từ hình ảnh cung cấp:

Kết quả đầu ra là hình ảnh đã được xử lý:

Các khung bao được vẽ quanh người, xe hơi, đèn giao thông, túi xách.

Hiển thị số lượng:

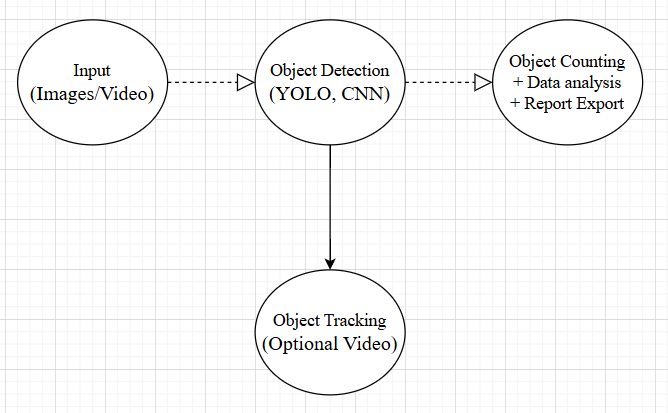
**Person (người):** 10

**Car (xe hơi):** 5

**Traffic light (đèn giao thông):** 2

**Handbag (túi xách):** 3

## **2.2 Xây dựng hệ thống**

****

*Hình 2.2 Sơ đồ YOLO và CNN*

### **2.2.1 Mô tả các bước trong sơ đồ:**

**Input (Đầu vào):**

Người dùng tải ảnh hoặc video lên hệ thống qua giao diện.

Hỗ trợ các định dạng ảnh phổ biến (JPEG, PNG) và video (MP4, AVI).

**Hệ thống nhận dạng đối tượng:**

**YOLO (You Only Look Once):**

Thuật toán phát hiện nhanh và chính xác các đối tượng trong ảnh/video, cho kết quả với khung bao (bounding box).

**CNN (Convolutional Neural Network):**

Được sử dụng để phân loại và nhận diện chính xác các loại đối tượng trong ảnh/video.

**Object Counting (Đếm đối tượng):**

Sau khi phát hiện, hệ thống tính toán số lượng từng loại đối tượng được nhận dạng.

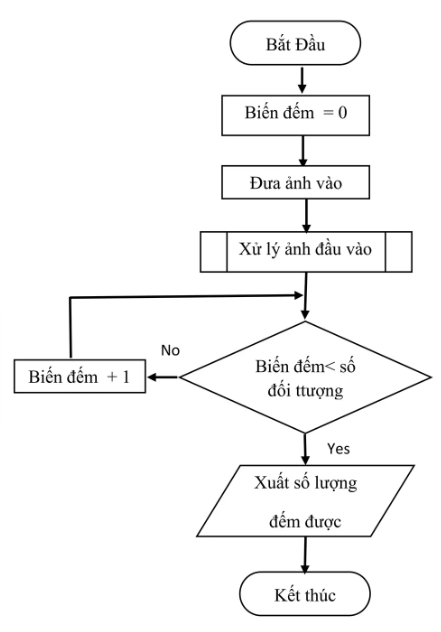
Tích hợp phân tích dữ liệu và xuất báo cáo chi tiết (số lượng, tỷ lệ các loại đối tượng).

**Object Tracking (Tùy chọn):**

Nếu xử lý video, hệ thống bổ sung module theo dõi đối tượng qua các khung hình.

Hỗ trợ thống kê tổng số lượng và vị trí đối tượng theo thời gian.

### **2.2.2 Sơ đồ đếm đối tượng**



*Hình 2.2 Sơ đồ đếm đối tượng*

# **CHƯƠNG III. KẾT QUẢ THỰC NGHIỆM**

## **3.1 Dữ liệu**

### **3.1.1 Nguồn Dữ liệu**

**Dataset**: Chúng tôi sử dụng các bộ dữ liệu công khai nhằm cung cấp một tập hợp phong phú các hình ảnh cho việc huấn luyện và kiểm tra mô hình. Một số bộ dữ liệu nổi bật bao gồm:

**COCO (Common Objects in Context)**: Bao gồm hơn 330.000 hình ảnh với 80 loại đối tượng khác nhau, mỗi hình ảnh có thể chứa nhiều đối tượng. Dữ liệu được gán nhãn chi tiết với bounding box, phân khúc và mô tả.

**PASCAL VOC**: Tập dữ liệu này có các hình ảnh được gán nhãn cho 20 loại đối tượng, với các bài toán phân loại, phát hiện và phân khúc.

**Bộ dữ liệu tự tạo**: Chúng tôi cũng xây dựng bộ dữ liệu riêng từ các hình ảnh thực tế, bao gồm các đối tượng trong các ngữ cảnh khác nhau (ví dụ: các loại xe, người, động vật) và được gán nhãn bằng tay để đảm bảo tính chính xác.

### **Tiền xử lý Dữ liệu**

**Chuẩn hóa kích thước**: Để đảm bảo rằng các hình ảnh đầu vào có kích thước đồng nhất, tất cả hình ảnh được thay đổi kích thước về 224x224 pixel. Việc này giúp mô hình dễ dàng xử lý, giảm thiểu độ phức tạp tính toán.

**Augmentation (Tăng cường dữ liệu)**: Chúng tôi áp dụng nhiều kỹ thuật augmentation như:

**Quay (Rotation)**: Quay hình ảnh với các góc khác nhau (0°, 90°, 180°, 270°) để giúp mô hình học được các góc nhìn khác nhau của đối tượng.

**Lật (Flipping)**: Lật ngang và lật dọc hình ảnh, hữu ích trong các bài toán mà đối tượng có thể xuất hiện từ cả hai hướng.

**Thay đổi độ sáng (Brightness Adjustment)**: Thay đổi độ sáng và độ tương phản của hình ảnh để mô hình có thể hoạt động tốt hơn trong các điều kiện ánh sáng khác nhau.

**Cắt ngẫu nhiên (Random Cropping)**: Cắt một phần ngẫu nhiên của hình ảnh để tạo ra sự đa dạng cho dữ liệu huấn luyện.

## **3.2 Các độ đo**

### **3.2.1. Độ Chính Xác (Accuracy)**

**Công thức tính**

Độ chính xác được xác định bằng công thức:

A black text on a white background

Description automatically generated

Trong đó:

**Số lượng dự đoán đúng** là tổng số lần mô hình phân loại hoặc nhận dạng chính xác các đối tượng so với nhãn thực tế.

**Tổng số dự đoán** là tổng tất cả các trường hợp (bao gồm cả dự đoán đúng và sai).

Ví dụ: Nếu mô hình phát hiện và phân loại 100 đối tượng trong ảnh, trong đó có 90 đối tượng được phân loại đúng, thì độ chính xác là:

A math problem with numbers

Description automatically generated with medium confidence

**Ý nghĩa:**

Độ chính xác là một chỉ số cơ bản để đánh giá hiệu suất của mô hình. Một độ chính xác cao thường cho thấy mô hình có khả năng phân loại hoặc nhận dạng đối tượng trong ảnh một cách hiệu quả. Tuy nhiên, chỉ sử dụng độ chính xác để đánh giá có thể không đủ, đặc biệt đối với các bài toán có sự mất cân bằng giữa các lớp dữ liệu (class imbalance).

Ví dụ:

Trong một bài toán phát hiện bệnh trong y tế, nếu chỉ 1% bệnh nhân thực sự mắc bệnh và mô hình luôn dự đoán "không mắc bệnh", độ chính xác sẽ rất cao (~99%) nhưng hiệu quả thực tế của mô hình lại rất thấp. Điều này do mô hình không phát hiện được bệnh nhân mắc bệnh (sai lầm nghiêm trọng trong thực tế).

**Tầm quan trọng của việc kết hợp độ chính xác với các chỉ số khác**

Để đánh giá mô hình một cách toàn diện hơn, cần kết hợp độ chính xác với các chỉ số khác, chẳng hạn như:

**Precision (Độ chính xác của từng lớp):** Xác định tỷ lệ dự đoán đúng trên tổng số các dự đoán được thực hiện cho một lớp cụ thể.

**Recall (Độ nhạy):** Đánh giá khả năng phát hiện đúng các đối tượng thực sự thuộc về một lớp cụ thể.

**F1-Score:** Là trung bình điều hòa giữa Precision và Recall, đặc biệt hữu ích trong trường hợp dữ liệu mất cân bằng.

**Ứng dụng trong bài toán phát hiện đối tượng**

Trong lĩnh vực phát hiện đối tượng trong ảnh, độ chính xác cao cho thấy mô hình có khả năng nhận diện và phân loại đúng hầu hết các đối tượng. Tuy nhiên, để đảm bảo mô hình hoạt động hiệu quả trong thực tế, các chỉ số như **IoU (Intersection over Union)** hoặc **Mean Average Precision (mAP)** thường được sử dụng cùng với độ chính xác để cung cấp đánh giá toàn diện hơn.

**Kết luận:**

Độ chính xác là một thước đo quan trọng và dễ hiểu trong đánh giá hiệu suất mô hình. Tuy nhiên, không nên phụ thuộc hoàn toàn vào chỉ số này, đặc biệt trong các bài toán yêu cầu độ chính xác cao hoặc đối mặt với dữ liệu không cân bằng. Việc kết hợp nhiều chỉ số đánh giá sẽ giúp tối ưu hóa và cải thiện mô hình một cách toàn diện hơn.

### **3.2.2.** **Precision và Recall**

**Precision (Độ chính xác của dự đoán)**

**Công thức**

A math equation with black text

Description automatically generated

**Trong đó:**

**TP (True Positive):** Số lượng đối tượng được dự đoán đúng (dự đoán là "đúng" và thực tế cũng là "đúng").

**FP (False Positive):** Số lượng đối tượng được dự đoán sai (dự đoán là "đúng" nhưng thực tế lại "sai").

**Ý nghĩa:**

Precision thể hiện độ tin cậy của các dự đoán mà mô hình thực hiện. Một giá trị Precision cao có nghĩa là trong tất cả các dự đoán "đúng" mà mô hình đưa ra, phần lớn thực sự là đúng trong thực tế. Precision đặc biệt quan trọng trong các trường hợp mà chi phí của một dự đoán sai là rất lớn.

**Ví dụ:**

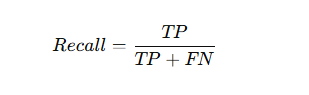
Trong một hệ thống phát hiện gian lận ngân hàng, nếu Precision thấp, mô hình sẽ gắn nhãn nhiều giao dịch hợp pháp là gian lận, gây phiền toái cho khách hàng. Precision cao sẽ giúp giảm thiểu các cảnh báo sai (false alarms).

Trong phát hiện đối tượng, Precision cao cho thấy mô hình không chỉ nhận diện được nhiều đối tượng mà còn rất ít gắn nhãn sai cho các vùng không phải đối tượng.

Precision cũng liên quan đến việc giảm thiểu các cảnh báo sai, giúp cải thiện độ chính xác trong các hệ thống nhạy cảm như y tế hoặc an ninh.

**Recall (Độ bao phủ hoặc độ nhạy):**

**Công thức:**



**Trong đó:**

**TP (True Positive):** Số lượng đối tượng được dự đoán đúng.

**FN (False Negative):** Số lượng đối tượng thực tế nhưng mô hình không phát hiện được (bỏ sót).

**Ý nghĩa:**

Recall đo lường khả năng của mô hình trong việc phát hiện ra các đối tượng thực tế trong dữ liệu. Một giá trị Recall cao đồng nghĩa với việc mô hình có khả năng nhận diện gần như tất cả các đối tượng cần phát hiện. Tuy nhiên, điều này không đảm bảo rằng các dự đoán đều chính xác (vì Precision có thể thấp).

### **3.2.3. F1 Score**

**Công thức**

**A black text with a black line

Description automatically generated**

**Trong đó:**

**Precision** (Độ chính xác): Đo lường tỷ lệ các dự đoán đúng trong tổng số các dự đoán được gắn nhãn là đúng.

**Recall** (Độ nhạy): Đo lường khả năng phát hiện chính xác các đối tượng thực sự.

F1 Score là trung bình điều hòa giữa Precision và Recall, giúp cân bằng hai chỉ số này và tạo ra một giá trị tổng hợp duy nhất để đánh giá hiệu suất của mô hình.

**Ý nghĩa:**

F1 Score cung cấp một cái nhìn toàn diện về hiệu suất của mô hình trong các bài toán phân loại, đặc biệt là khi Precision và Recall có giá trị khác biệt nhau.

Nếu F1 Score cao, điều đó cho thấy mô hình vừa có độ chính xác cao (Precision cao) vừa có khả năng phát hiện tốt các đối tượng thực tế (Recall cao).

Nếu F1 Score thấp, mô hình hoặc có nhiều dự đoán sai (Precision thấp), hoặc bỏ sót nhiều đối tượng (Recall thấp).

**F1 Score rất hữu ích trong các trường hợp sau:**

**Dữ liệu mất cân bằng giữa các lớp đối tượng:** Khi một lớp xuất hiện rất ít so với các lớp khác (ví dụ, trong phát hiện gian lận hoặc phát hiện bệnh hiếm gặp), chỉ số Accuracy (Độ chính xác tổng thể) thường không phản ánh đúng hiệu suất của mô hình. F1 Score giúp đánh giá hiệu suất chi tiết hơn bằng cách cân bằng giữa Precision và Recall.

**Yêu cầu độ chính xác cao cho cả hai chỉ số:** Trong nhiều ứng dụng, việc chỉ tập trung tối ưu hóa Precision hoặc Recall không đủ, và cần một chỉ số tổng hợp để đánh giá toàn diện.

### **3.2.4. Mean Average Precision (mAP)**

**Cách tính**

Mean Average Precision (mAP) là một chỉ số quan trọng trong các bài toán phát hiện đối tượng (object detection), giúp đánh giá hiệu suất của mô hình dựa trên cả **Precision** và **Recall**. mAP được tính toán thông qua các bước sau:

**Tính chỉ số IoU (Intersection over Union):**

IoU đo lường mức độ trùng khớp giữa vùng dự đoán (predicted bounding box) và vùng thực tế (ground truth bounding box).

**Công thức:**

A black text with black text

Description automatically generated with medium confidence

IoU cao (gần 1) cho thấy vùng dự đoán rất gần với vùng thực tế.

**Xác định ngưỡng IoU:**

Một ngưỡng IoU cụ thể (ví dụ: 0.5 hoặc 0.75) được sử dụng để quyết định một dự đoán có được coi là đúng (True Positive) hay không.

**Tính Average Precision (AP) cho từng lớp:**

AP được tính bằng cách xác định diện tích dưới đường Precision-Recall (PR) Curve cho một lớp đối tượng cụ thể.

PR Curve được xây dựng bằng cách sắp xếp các dự đoán theo độ tin cậy (confidence score), sau đó tính Precision và Recall tại mỗi mức độ tin cậy.

**Tính Mean Average Precision (mAP):**

mAP là giá trị trung bình của các AP trên tất cả các lớp đối tượng. Nếu có nnn lớp, mAP được tính như sau:

A mathematical equation with numbers and symbols

Description automatically generated

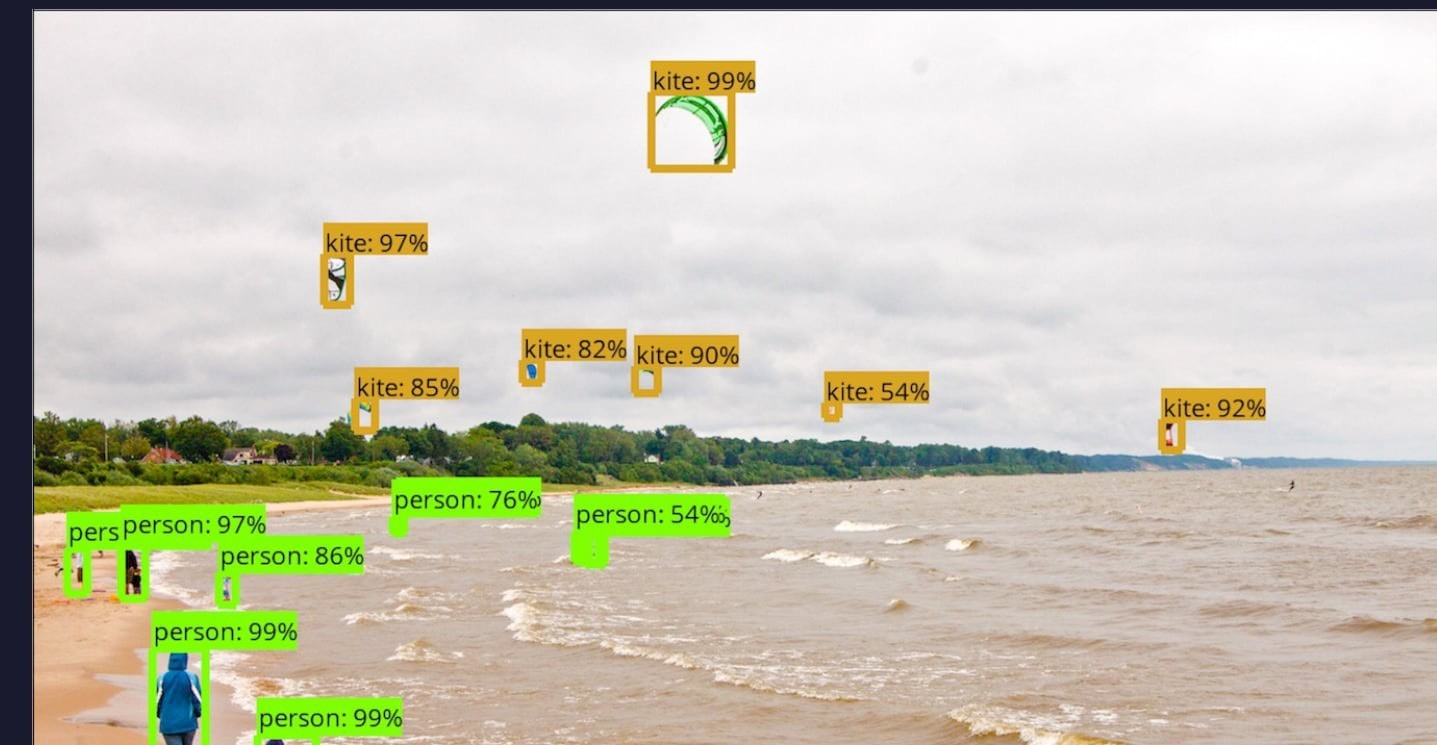
**Ý nghĩa:**

**Đánh giá toàn diện hiệu suất của mô hình:**  
mAP không chỉ đánh giá độ chính xác trong phát hiện đối tượng mà còn cân nhắc khả năng bao phủ các đối tượng thực tế qua các ngưỡng IoU khác nhau.

**Đánh giá theo từng lớp đối tượng:**  
Bằng cách tính AP cho từng lớp và trung bình hóa, mAP cung cấp một chỉ số tổng hợp phản ánh khả năng phát hiện đối tượng trên nhiều loại đối tượng khác nhau. Điều này rất quan trọng trong các bài toán có dữ liệu đa dạng, ví dụ: phát hiện nhiều loại động vật trong ảnh tự nhiên.

**Tương thích với nhiều bài toán:**  
mAP không chỉ hữu ích trong phát hiện đối tượng mà còn được áp dụng rộng rãi trong các lĩnh vực khác như phân loại hình ảnh hoặc nhận diện khuôn mặt, nơi cần đánh giá mô hình trên nhiều lớp hoặc nhiều ngưỡng độ tin cậy.

## **3.3** **Kết quả thực nghiệm**

*****Hình 3.1 Kết quả nhận diện*

# **KẾT LUẬN**

**Kết quả đạt được**

**Phân tích hệ thống**

Chúng em đã tiến hành phân tích hệ thống một cách chi tiết, xác định rõ các yêu cầu chức năng và phi chức năng của ứng dụng nhận diện và đếm đối tượng. Qua quá trình nghiên cứu và khảo sát, hệ thống đã được thiết kế nhằm đáp ứng nhu cầu nhận diện và đếm đối tượng một cách chính xác và nhanh chóng, đặc biệt trong các ứng dụng như giám sát giao thông,các hệ thống an ninh tự động,…

**Đáp ứng yêu cầu**

Ứng dụng đã hoàn thành tốt các yêu cầu cơ bản, bao gồm khả năng nhận diện và đếm đối tượng trong các điều kiện khác nhau về ánh sáng, góc chụp và tốc độ di chuyển. Các chức năng như nhận diện, phân tích và hiển thị kết quả đã hoạt động hiệu quả. Giao diện người dùng của hệ thống đơn giản và dễ sử dụng, giúp người dùng có thể tương tác một cách nhanh chóng và trực quan.

**Thiết kế giao diện và trải nghiệm người dùng**

Giao diện của ứng dụng được thiết kế với tiêu chí đơn giản và dễ sử dụng, nhằm đáp ứng nhu cầu của người sử dụng trong môi trường thực tế. Các chức năng chính như nhận diện và đếm, hiển thị kết quả và quản lý dữ liệu đều được bố trí một cách hợp lý, thuận tiện cho người dùng. Trải nghiệm người dùng được chú trọng, giúp giảm thiểu sai sót trong quá trình thao tác và nâng cao hiệu quả công việc.

**Những hạn chế**

Độ chính xác trong điều kiện khó khăn: Hệ thống có thể gặp khó khăn trong việc nhận diện biển số khi điều kiện ánh sáng yếu, đối tượng mờ và di chuyển nhanh.

Khả năng nhận diện hạn chế: Hệ thống hiện chỉ hỗ trợ nhận diện và đếm trong một số khu vực cụ thể và có thể không phù hợp với nhiều khu vực khác nhau.

Tốc độ xử lý: Mặc dù đã được tối ưu hóa, hệ thống vẫn có thể gặp vấn đề khi xử lý số lượng lớn đối tượng trong thời gian thực, đặc biệt trong các tình huống đối tượng bị mờ, che khuất một phần.

Yêu cầu tài nguyên phần cứng: Việc áp dụng các thuật toán phức tạp như học sâu có thể yêu cầu phần cứng mạnh mẽ và tài nguyên tính toán cao, điều này có thể gây khó khăn khi triển khai trên các thiết bị có cấu hình thấp.

Bảo mật dữ liệu: Việc thu thập và lưu trữ thông tin đối tượng có thể tiềm ẩn các vấn đề về bảo mật và quyền riêng tư nếu không được xử lý đúng cách.

**Hướng phát triển trong tương lai**

Cải thiện độ chính xác nhận diện: Nâng cao khả năng nhận diện và đếm đối tượng trong các điều kiện khó khăn bằng cách áp dụng công nghệ học sâu (Deep Learning).

Tích hợp vào ứng dụng thực tế: Ứng dụng trong các hệ thống giám sát giao thông, bãi đỗ xe, siêu thi,….

Xử lý thời gian thực: Tối ưu hóa thuật toán để nâng cao khả năng xử lý nhanh trong môi trường thực tế.

Tăng cường bảo mật: Đảm bảo bảo mật và quyền riêng tư cho người sử dụng thông qua mã hóa và bảo vệ dữ liệu.

# **TÀI LIỆU THAM KHẢO**

1. A. Zelinsky, *Learning OpenCV---Computer Vision with the OpenCV Library (Bradski, G.R. et al.; 2008)[On the Shelf]*, vol. 16, no. 3. 2009.
2. Chris Dahms (2016), OpenCV 3 License Plate Recognition Python. <https://www.youtube.com/watch?v=fJcl6Gw1D8k>
3. OpenCV. Morphological Transformations.<https://docs.opencv.org/3.4/d9/d61/tutorial_py_morphological_ops.html>
4. "Pattern Recognition and Machine Learning" - Christopher Bishop
5. Kaggle: Image Classification Datasets.

https://www.kaggle.com