**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 1: Xây dựng hệ thống phát hiện đối tượng và trích xuất thông tin đối tượng trong ảnh**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Mã sinh viên** | **Sinh viên thực hiện** | **Lớp hành chính** |
| **1** | 20210773 | Nguyễn Trọng Quỳnh | DCCNTT12.10.3 |
| **2** | 20210850 | Nguyễn Thế Bảo | DCCNTT12.10.3 |
| **3** | 20210827 | Đỗ Tuấn Anh | DCCNTT12.10.3 |
| **4** | 20210836 | Nguyễn Hoàng Nam | DCCNTT12.10.3 |
| **5** | 20210662 | Vũ Minh Hoàn | DCCNTT12.10.3 |

**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 1: Xây dựng hệ thống phát hiện đối tượng và trích xuất thông tin đối tượng trong ảnh**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Mã sinh viên** | **Sinh viên thực hiện** | **Lớp hành chính** |
| **1** | 20210773 | Nguyễn Trọng Quỳnh | DCCNTT12.10.3 |
| **2** | 20210850 | Nguyễn Thế Bảo | DCCNTT12.10.3 |
| **3** | 20210827 | Đỗ Tuấn Anh | DCCNTT12.10.3 |
| **4** | 20210836 | Nguyễn Hoàng Nam | DCCNTT12.10.3 |
| **5** | 20210662 | Vũ Minh Hoàn | DCCNTT12.10.3 |

|  |  |
| --- | --- |
|  |  |

**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: 1**  **Tên học phần: Xử Lý Ảnh và Thị Giác Máy Tính**  **Lớp Tín chỉ: XATGMT.03.K12.03.LH.C04.1\_LT** | |
| **Cán bộ chấm thi 1**  *(Ký và ghi rõ họ tên)* | **Cán bộ chấm thi 2**  *(Ký và ghi rõ họ tên)* |

| **TT** | **TIÊU CHÍ** | **THANG ĐIỂM** | **TÊN SV 1** | **TÊN SV 2** | **TÊN SV 3** | **TÊN SV 4** | **TÊN SV 5** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| MÃ SV 1 | MÃ SV 2 | MÃ SV 3 | MÃ SV 4 | MÃ SV 5 |
| **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* |  |  |  |  |  |

**LỜI MỞ ĐẦU**

Khoa học công nghệ ngày càng phát triển mạnh mẽ trong bối cảnh cạnh tranh giữa các công ty và sự bùng nổ của nhiều công nghệ hiện đại. Các từ khóa như IoT (Internet vạn vật), trí tuệ nhân tạo (AI), điện toán đám mây (Cloud Computing), chuỗi khối (Blockchain), dữ liệu lớn (Big Data), trải nghiệm thực tế ảo (Virtual Reality) và thực tế tăng cường (Augmented Reality) ngày càng nhận được sự quan tâm sâu rộng. Những công nghệ này không hoạt động riêng lẻ mà thường kết hợp và hỗ trợ lẫn nhau để giải quyết các bài toán thực tiễn một cách hiệu quả và tối ưu.

Trong bối cảnh đó, việc sử dụng công nghệ để xử lý hình ảnh đã đạt được nhiều bước tiến vượt bậc. Các kỹ thuật nhận dạng đối tượng, học máy và học sâu giúp việc phân tích hình ảnh, trích xuất thông tin hoặc cung cấp các chú thích về đối tượng trong ảnh trở nên nhanh chóng và chính xác hơn. Những tiến bộ này không chỉ mở ra cơ hội giải quyết nhiều bài toán phức tạp trong thực tế mà còn cải thiện hiệu quả ứng dụng công nghệ trong các lĩnh vực như giám sát an ninh, quản lý dữ liệu, y tế, giao thông và thương mại điện tử.

Đề tài “Xây dựng hệ thống phát hiện đối tượng và trích xuất thông tin đối tượng trong ảnh” tập trung nghiên cứu và ứng dụng các phương pháp hiện đại để phát hiện đối tượng trong hình ảnh, sau đó trích xuất thông tin liên quan đến các đối tượng đó. Mục tiêu chính là xây dựng một hệ thống tự động, thân thiện với người dùng, có khả năng hỗ trợ tốt trong việc quản lý và xử lý dữ liệu hình ảnh.

Hệ thống sẽ bao gồm các bước như: tiền xử lý ảnh, phát hiện đối tượng thông qua các thuật toán học máy hoặc học sâu, và trích xuất thông tin từ các đối tượng được phát hiện. Kết quả từ đề tài không chỉ giúp nâng cao kiến thức về xử lý ảnh và thị giác máy tính mà còn hướng tới việc áp dụng vào các tình huống thực tế, giải quyết bài toán nhận dạng và xử lý thông tin từ hình ảnh một cách hiệu quả.

**Mục Lục**

[**DANH SÁCH HÌNH ẢNH** 9](#_Toc184457698)

[**CHƯƠNG I: GIỚI THIỆU** 10](#_Toc184457699)

[**1.1 Thế nào là nhận dạng đối tượng?** 10](#_Toc184457700)

[**1.1.2. Các thành phần chính của nhận dạng đối tượng.** 10](#_Toc184457701)

[**1.2. Các kĩ thuật sử dụng trong bài toán nhận dạng** 11](#_Toc184457702)

[**1.2.1: Phát hiện khuôn mặt dựa trên Haar Cascades** 11](#_Toc184457703)

[**1.2.2. HOG (Histogram of Oriented Gradients)** 12](#_Toc184457704)

[**1.2.3. Mạng nơ-ron tích chập (CNN)** 12](#_Toc184457705)

[**1.2.4. DeepFace và các thư viện tích hợp** 12](#_Toc184457706)

[**1.3. Ngôn ngữ lập trình và thư viện sử dụng** 13](#_Toc184457707)

[**1.3.1. Python** 13](#_Toc184457708)

[**1.3.2. Python mang lại những lợi ích gì?** 13](#_Toc184457709)

[**1.3.3. Khoa học dữ liệu và máy học** 14](#_Toc184457710)

[**1.3.4. Python có những đặc điểm gì?** 14](#_Toc184457711)

[**1.3.5 Thư viện Python là gì?** 15](#_Toc184457712)

[**1.3.6. Ưu điểm và nhược điểm của Python** 15](#_Toc184457713)

[**CHƯƠNG II: XÂY DỰNG CHƯƠNG TRÌNH** 18](#_Toc184457714)

[**2.1 Mô tả bài toán** 18](#_Toc184457715)

[**2.2 Xây dựng hệ thống** 18](#_Toc184457716)

[**2.2.1 Thu Thập dữ liệu** 19](#_Toc184457717)

[**2.2.2 Tiền xử lý hình ảnh** 21](#_Toc184457718)

[**2.2.3. Trích xuất đặc trưng** 22](#_Toc184457719)

[**2.2.4 Nhận dạng đối tượng** 23](#_Toc184457720)

[**2.2.5. Truy xuất thông tin** 24](#_Toc184457721)

[**2.2.6. Mô hình nhận diện khuôn mặt (Face Recognition Model)** 25](#_Toc184457722)

[**2.2.7 Thuật toán CNN** 27](#_Toc184457723)

[**2.2 Các giai đoạn hệ thông** 28](#_Toc184457724)

[**2.2.1. Giai đoạn quản lý dữ liệu** 28](#_Toc184457725)

[**2.2.2. Giai đoạn huận luyện mô hình học máy** 28](#_Toc184457726)

[**2.2.3. Giai đoạn nhận dạng đối tượng** 29](#_Toc184457727)

[**2.2.4. Giai đoạn trích xuất thông tin** 30](#_Toc184457728)

[**2.3 Xây dựng hệ thống** 31](#_Toc184457729)

[**2.3.1 Các điều kiện để xây dựng hệ thống** 31](#_Toc184457730)

[**2.3.2 Tổ chức và quản lý dự án** 32](#_Toc184457731)

[**CHƯƠNG III: THỰC NGHIỆM KẾT QUẢ** 34](#_Toc184457732)

[**3.1. Ngôn ngữ lập trình và thư viện sử dụng** 34](#_Toc184457733)

[**3.1.1. Python** 34](#_Toc184457734)

[**3.1.2 Những thư viện trong học máy được sử dụng trong bài toán.** 34](#_Toc184457735)

[**3.2. Các bước chạy chương trình** 37](#_Toc184457736)

[**3.3. Dữ liệu** 39](#_Toc184457737)

[**3.3.1 Mô tả dữ liệu đầu vào** 40](#_Toc184457738)

[**3.2.2 Xử lý dữ liệu** 40](#_Toc184457739)

[**Tài liệu tham khảo** 46](#_Toc184457740)

# **DANH SÁCH HÌNH ẢNH**

Hình 2.1 Sơ đồ tổng quát nhận dạng đối tượng trong ảnh 19

Hình 2.1.2 Các bước trong nhận diện đối tượng 26

Hình 3.1.1 Hình ảnh đối tượng 37

Hình 3.1.2: HìnhTạo encoding từ ảnh khuôn mặt 38

Hình 3.1.3 Ảnh đầu ra sau khi nhận diện 39

Hình 3.3.1 Hình ảnh dữ liệu của các đối tượng cần nhận dạng 40

Hình 3.3.1 kết quả nhận diện đối tượng 42

Hình 3.3.2 Xuất thông tin ra Teminal 43

Hình 3.3.3 Xuất thông tin ra file output.txt 43

# **CHƯƠNG I: GIỚI THIỆU**

# **1.1 Thế nào là nhận dạng đối tượng?**

Nhận dạng đối tượng (Object recognition) là một lĩnh vực trong thị giác máy tính nhằm xác định, phân loại và nhận diện các đối tượng trong ảnh hoặc video. Đây là một bước quan trọng để máy tính hiểu được nội dung của hình ảnh, giúp hệ thống tự động phân tích và xử lý thông tin thị giác.

## **1.1.2. Các thành phần chính của nhận dạng đối tượng.**

Các thành phần chính của nhận dạng đối tượng

**Phát hiện đối tượng (Object Detection):** Xác định vị trí của các đối tượng trong ảnh bằng cách tạo ra các bounding boxes xung quanh chúng.

**Phân loại đối tượng (Object Classification):** Xác định đối tượng trong ảnh thuộc nhóm hoặc lớp nào, ví dụ: "người", "xe hơi", "mèo", "cây".

**Nhận diện đối tượng (Object Identification):** Không chỉ phân loại mà còn nhận diện danh tính cụ thể của đối tượng, ví dụ: "Đây là con mèo tên Luna" thay vì chỉ nhận biết "mèo".

**Phân đoạn đối tượng (Object Segmentation):**Phân chia từng đối tượng trong ảnh với biên dạng chính xác, thay vì chỉ sử dụng bounding boxes (ví dụ: Phân đoạn hình ảnh của một con người đến từng chi tiết như tay, chân).

Quy trình thực hiện nhận dạng đối tượng

**Thu thập dữ liệu:**

Thu thập tập hợp ảnh hoặc video chứa các đối tượng cần nhận dạng.

**Gắn nhãn dữ liệu (Data Annotation):**

Đánh dấu vùng đối tượng và gán nhãn cho từng loại đối tượng.

**Huấn luyện mô hình:**

Sử dụng dữ liệu đã gắn nhãn để huấn luyện các mô hình học sâu (Deep Learning), ví dụ: YOLO, SSD, hoặc Faster R-CNN.

**Dự đoán:**

Đưa hình ảnh mới vào mô hình đã huấn luyện để phát hiện, phân loại và nhận diện đối tượng.

Ứng dụng của nhận dạng đối tượng

1. An ninh và giám sát:
   * Phát hiện và theo dõi người hoặc phương tiện trong camera an ninh.
   * Nhận dạng khuôn mặt trong các hệ thống bảo mật.
2. Giao thông:
   * Nhận dạng biển báo giao thông, xe cộ hoặc người đi bộ.
3. Y tế:
   * Phát hiện các tổn thương, khối u hoặc dấu hiệu bệnh trong ảnh y khoa.
4. Thương mại điện tử:
   * Nhận dạng sản phẩm từ hình ảnh để quản lý hoặc tìm kiếm nhanh.
5. Robot và xe tự hành:
   * Phát hiện và tránh vật cản trong môi trường thực tế.

Các bước cơ bản trong bài toán nhận dạng đối tượng trong ảnh

**Thu thâp và xử lý dữ liệu.**

* Thu thập hình ảnh: Tập dữ liệu ảnh được lấy từ kho ảnh của máy điện thoại bao gồm các đối tượng cần nhận dạng.
* Tiền xử lý ảnh: Chuẩn hóa kích thước, loại bỏ nhiễu, gắn nhãn (xác định các bounding boxes và các lớp đối tượng.

**Xây dựng mô hình nhận dạng**

* Chọn mô hình: Dựa trên mức độ phức tạp bài toán và tài nguyên, chọn mô hình phù hợp.
* Huấn luyện: Sử dụng tệp dữ liệu đã chuẩn bị để huấn luyện mô hình.

**Phát hiện và phân loại đối tượng**

* Yêu cầu: Phát hiện: xác định vị trí của đối tượng trong ảnh (bounding box)

Phân loại: xác định đối tượng thuộc lớp nào (classification)

 Vấn đề: Đảm bảo phát hiện chính xác các đối tượng trong từng trường hợp:

* Nhiều đối tượng chồng lấn.
* Đối tượng nhỏ hoặc bị che khuất một phần.
* Tăng tốc độ xử lý đối với ứng dụng thời gian thực

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

Với mỗi kĩ thuật gồm: tư tưởng kĩ thuật, hình mô tả kĩ thuật và ưu nhược điểm

## **1.2.1: Phát hiện khuôn mặt dựa trên Haar Cascades**

**Tư tưởng kỹ thuật:**

* Sử dụng các đặc trưng Haar (Haar Features), tương tự như các bộ lọc, để xác định các đặc điểm khuôn mặt như mắt, miệng, mũi.
* Bộ phân loại cascade được huấn luyện bằng cách sử dụng nhiều cửa sổ ảnh khác nhau để phát hiện khuôn mặt ở nhiều kích thước.

**Mô tả kỹ thuật:**

* Input: Hình ảnh.
* Output: Các bounding box chứa khuôn mặt được phát hiện.

**Ưu điểm:**

* Dễ triển khai và nhanh chóng cho phát hiện thời gian thực.
* Hiệu quả trên các thiết bị phần cứng thấp.

**Nhược điểm:**

* Không hiệu quả với các góc nhìn khác nhau hoặc hình ảnh có độ sáng không đồng đều.
* Phụ thuộc vào dữ liệu huấn luyện.

## **1.2.2. HOG (Histogram of Oriented Gradients)**

**Tư tưởng kỹ thuật:**

* Trích xuất các đặc trưng gradient (gradient features) từ hình ảnh để biểu diễn biên và các đường cong của khuôn mặt.
* Sử dụng SVM (Support Vector Machine) để phân loại.

**Mô tả kỹ thuật:**

* Input: Hình ảnh.
* Output: Các bounding box chứa khuôn mặt được phát hiện.

**Ưu điểm:**

* Chính xác cao với dữ liệu chuẩn.
* Không yêu cầu dữ liệu huấn luyện lớn.

**Nhược điểm:**

* Chậm hơn so với Haar Cascades.
* Không hiệu quả với các góc xoay lớn hoặc khuôn mặt nhỏ.

## **1.2.3. Mạng nơ-ron tích chập (CNN)**

**Tư tưởng kỹ thuật:**

* Sử dụng các mạng sâu (deep neural networks) để học các đặc trưng từ dữ liệu.
* Phân loại hoặc phát hiện khuôn mặt dựa trên các lớp tích chập (convolutional layers).

**Mô tả kỹ thuật:**

* Input: Hình ảnh.
* Output: Vị trí khuôn mặt và đặc trưng.

**Ưu điểm:**

* Chính xác cao.
* Hiệu quả với các dữ liệu phức tạp, bao gồm góc xoay và độ sáng khác nhau.

**Nhược điểm:**

* Cần lượng dữ liệu lớn và thời gian huấn luyện lâu.
* Yêu cầu phần cứng mạnh.
* Độ phức tạp và khó khăn trong đào tạo
* Thiếu khả năng diễn giải

## **1.2.4. DeepFace và các thư viện tích hợp**

**Tư tưởng kỹ thuật:**

* Sử dụng thư viện như DeepFace, kết hợp các phương pháp tiên tiến như VGG-Face, FaceNet, OpenFace, và DeepID để nhận dạng khuôn mặt.

**Mô tả kỹ thuật:**

* Input: Hình ảnh.
* Output: Xác định danh tính hoặc so khớp khuôn mặt.

**Ưu điểm:**

* Dễ triển khai với các thư viện mạnh mẽ.
* Hỗ trợ nhiều thuật toán trong một nền tảng.

**Nhược điểm:**

* Đôi khi cần tinh chỉnh cấu hình.
* Yêu cầu cài đặt phụ thuộc.

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

## **1.3.1. 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). Các nhà phát triển sử dụng Python vì nó hiệu quả, dễ học và có thể chạy trên nhiều nền tảng khác nhau. Phần mềm Python được tải xuống miễn phí, tích hợp tốt với tất cả các loại hệ thống và tăng tốc độ phát triển.

## **1.3.2. Python mang lại những lợi ích gì?**

Những lợi ích của Python bao gồm:

* Các nhà phát triển có thể dễ dàng đọc và hiểu một chương trình Python vì ngôn ngữ này có cú pháp cơ bản giống tiếng Anh.
* Python giúp cải thiện năng suất làm việc của các nhà phát triển vì so với những ngôn ngữ khác, họ có thể sử dụng ít dòng mã hơn để viết một chương trình Python.
* Python có một thư viện tiêu chuẩn lớn, chứa nhiều dòng mã có thể tái sử dụng cho hầu hết mọi tác vụ. Nhờ đó, các nhà phát triển sẽ không cần phải viết mã từ đầu.
* Các nhà phát triển có thể dễ dàng sử dụng Python với các ngôn ngữ lập trình phổ biến khác như Java, C và C++.
* Cộng đồng Python tích cực hoạt động bao gồm hàng triệu nhà phát triển nhiệt tình hỗ trợ trên toàn thế giới. Nếu gặp phải vấn đề, bạn sẽ có thể nhận được sự hỗ trợ nhanh chóng từ cộng đồng.
* Trên Internet có rất nhiều tài nguyên hữu ích nếu bạn muốn học Python. Ví dụ: bạn có thể dễ dàng tìm thấy video, chỉ dẫn, tài liệu và hướng dẫn dành cho nhà phát triển.
* Python có thể được sử dụng trên nhiều hệ điều hành máy tính khác nhau, chẳng hạn như Windows, macOS, Linux và Unix.1.3.3. Tự động hóa bằng các tập lệnh Python

Ngôn ngữ tập lệnh là một ngôn ngữ lập trình tự động hóa các tác vụ mà thường được con người thực hiện. Các lập trình viên thường xuyên sử dụng các tập lệnh Python để tự động hóa nhiều tác vụ hàng ngày như:

* Đổi tên một số lượng lớn tệp cùng lúc
* Chuyển đổi một tệp sang một loại tệp khác
* Loại bỏ các từ trùng lặp trong tệp văn bản
* Thực hiện các phép tính toán cơ bản
* Gửi email
* Tải xuống nội dung
* Thực hiện phân tích nhật ký cơ bản
* Tìm kiếm lỗi trong nhiều tệp

## **1.3.3. Khoa học dữ liệu và máy học**

khoa học dữ liệu trích xuất thông tin quý giá từ dữ liệu và máy học (ML) dạy máy tính tự động học hỏi từ dữ liệu và đưa ra các dự đoán chính xác. Các nhà khoa học dữ liệu sử dụng Python cho các tác vụ khoa học dữ liệu sau:

* Sửa và loại bỏ dữ liệu không chính xác, hay còn được gọi là làm sạch dữ liệu
* Trích xuất và chọn lọc các đặc điểm đa dạng của dữ liệu
* Ghi nhãn dữ liệu gán tên có ý nghĩa cho dữ liệu
* Tìm các số liệu thống kê khác nhau từ dữ liệu
* Trực quan hóa dữ liệu bằng cách sử dụng các biểu đồ và đồ thị, chẳng hạn như biểu đồ đường, biểu đồ cột, biểu đồ tần suất và biểu đồ tròn

## **1.3.4. Python có những đặc điểm gì?**

**Các đặc điểm sau tạo nên sự độc đáo của ngôn ngữ lập trình Python:**

**Python là một ngôn ngữ thông dịch**

Python là một ngôn ngữ thông dịch, điều này nghĩa là ngôn ngữ này trực tiếp chạy từng dòng mã. Nếu có lỗi trong mã chương trình, nó sẽ ngừng chạy. Do đó, lập trình viên có thể nhanh chóng tìm ra lỗi trong đoạn mã.

**Python là một ngôn ngữ dễ sử dụng**

Python sử dụng từ ngữ giống trong tiếng Anh. Không giống như các ngôn ngữ lập trình khác, Python không sử dụng dấu ngoặc ôm. Thay vào đó, ngôn ngữ này sử dụng thụt đầu dòng.

**Python là một ngôn ngữ linh hoạt**

Các lập trình viên không cần phải khai báo loại biến khi viết mã bởi vì Python sẽ xác định chúng vào thời điểm chạy. Vì vậy, bạn có thể viết các chương trình Python một cách nhanh chóng hơn.

**Python là một ngôn ngữ cấp cao**

Python gần gũi với ngôn ngữ con người hơn các ngôn ngữ lập trình khác. Do đó, các lập trình viên không cần phải lo lắng về những chức năng cơ bản của nó như kiến trúc và quản lý bộ nhớ.

**Python là một ngôn ngữ lập trình hướng đối tượng**

Python coi mọi thứ đều là đối tượng, nhưng ngôn ngữ này cũng hỗ trợ các phương thức lập trình khác như lập trình hàm và lập trình cấu trúc.

## **1.3.5 Thư viện Python là gì?**

Thư viện là một tập hợp các mã thường xuyên được sử dụng mà các nhà phát triển có thể bao gồm trong những chương trình Python của họ để không phải lập trình từ đầu. Theo mặc định, Python đi kèm với Thư viện chuẩn, chứa rất nhiều các hàm có thể tái sử dụng. Ngoài ra, hơn 137.000 thư viện Python có sẵn cho các ứng dụng khác nhau, bao gồm phát triển web, khoa học dữ liệu và máy học (ML).

**Matplotlib**

Các nhà phát triển sử dụng Matplotlib để hiển thị dữ liệu dưới dạng đồ họa hai và ba chiều (2D và 3D) chất lượng cao. Thư viện này thường được sử dụng trong các ứng dụng khoa học. Với Matplotlib, bạn có thể trực quan hóa dữ liệu bằng cách hiển thị dữ liệu dưới dạng các biểu đồ khác nhau, chẳng hạn như biểu đồ cột và biểu đồ đường. Bạn cũng có thể hiển thị nhiều biểu đồ cùng lúc và các chi tiết đồ họa có thể được di chuyển qua mọi nền tảng.

**Pandas**

Pandas cung cấp cấu trúc dữ liệu được tối ưu hóa và linh hoạt mà bạn có thể sử dụng để thao tác với dữ liệu chuỗi thời gian và dữ liệu có cấu trúc, chẳng hạn như bảng và nhóm. Ví dụ, bạn có thể sử dụng Pandas để đọc, ghi, hợp nhất, lọc và nhóm dữ liệu. Thư viện này được nhiều người sử dụng cho các tác vụ khoa học dữ liệu, phân tích dữ liệu và ML.

**NumPy**

NumPy là một thư viện phổ biến mà các nhà phát triển sử dụng để dễ dàng tạo và quản lý nhóm, thao tác với các hình dạng logic và thực hiện các phép toán đại số tuyến tính. NumPy hỗ trợ tích hợp với nhiều ngôn ngữ như C và C ++.

**OpenCV-Python**

OpenCV-Python là một thư viện mà các nhà phát triển sử dụng để xử lý hình ảnh cho các ứng dụng thị giác máy tính. Thư viện này cung cấp nhiều hàm cho các tác vụ xử lý hình ảnh như đọc và ghi hình ảnh cùng lúc, xây dựng môi trường 3D từ môi trường 2D cũng như chụp và phân tích hình ảnh từ video.

**Keras**

Keras là thư viện mạng nơ-ron chuyên sâu của Python với khả năng hỗ trợ tuyệt vời cho việc xử lý dữ liệu, trực quan hóa và hơn thế nữa. Keras hỗ trợ nhiều mạng nơ-ron. Thư viện này có cấu trúc mô-đun mang lại sự linh hoạt cho việc lập trình các ứng dụng sáng tạo.

## **1.3.6. Ưu điểm và nhược điểm của Python**

**Ưu điểm Python**

Có một lý do mà các nhà phát triển chọn viết mã bằng Python. Nó có một số tính năng độc đáo giúp việc lập trình trở nên đơn giản hơn nhiều. Chúng ta hãy xem xét một số tính năng giúp làm việc với lợi thế của Python:

**Dễ đọc và dễ học**

Python là một ngôn ngữ đơn giản để đọc và học. Nó không có cú pháp phức tạp như các ngôn ngữ cấp cao khác như C hoặc C ++. Nhờ ít phức tạp hơn, Python cho phép bạn suy nghĩ rõ ràng hơn và tập trung vào việc xây dựng logic.

**Giảm chi phí bảo trì**

Do tính đơn giản của nó, Python giúp bảo trì ứng dụng dễ dàng hơn và do đó, giảm chi phí liên quan, đây là một lợi thế lớn.

**Tránh tác hại từ lỗi phần mềm**

Python không để lỗi mã xảy ra phân đoạn trong ứng dụng. Do đó, nó được coi là một ngôn ngữ ưa thích.

**Khả năng ứng dụng rộng rãi**

Một tính năng thiết yếu khác của ngôn ngữ này là nó có thể áp dụng rộng rãi. Các kỹ sư, nhà khoa học và nhà toán học sử dụng rộng rãi nó.

**Quản lý bộ nhớ**

Python có một thư viện rộng lớn với khả năng quản lý bộ nhớ, điều này làm cho nó nổi bật so với các ngôn ngữ lập trình khác. Nó bao gồm một heap riêng chứa tất cả các đối tượng và cấu trúc dữ liệu Python, một trình quản lý bộ nhớ tích hợp để duy trì heap riêng tư này

**Đơn giản và nhanh chóng**

Cộng đồng Python cung cấp hỗ trợ nhanh chóng và thiết thực cho người dùng cũng như khả năng thích ứng nhanh của mã. Một số chuyên gia thích đặt biệt danh cho Python là “ngôn ngữ sẵn sàng để chạy” vì nó chỉ yêu cầu mã đơn giản để được thực thi. Nâng cao và kiểm tra mã thoải mái hơn nhiều với Python.

**Mã hóa không đồng bộ**

Mã hóa không đồng bộ sử dụng một vòng lặp sự kiện duy nhất để hoàn thành công việc trong những khoảng thời gian nhỏ. Python rất hữu ích để viết mã không đồng bộ vì nó dễ viết và dễ bảo trì. Nó không yêu cầu bất kỳ nội dung nghiên cứu phức tạp, bế tắc hoặc bất kỳ sự phức tạp nào khác.

**Tích hợp với các ngôn ngữ khác**

Python có các thư viện như Cython và Jython, cho phép tích hợp với các ngôn ngữ khác như C, C ++ và Java để phát triển đa nền tảng. Đây là một trong những đặc quyền chính của Python vì không có ngôn ngữ nào là hoàn hảo và đôi khi sự phát triển đòi hỏi các chức năng ngôn ngữ đa dạng.

**Tích hợp ứng dụng doanh nghiệp**

Python là lựa chọn tốt nhất cho Tích hợp ứng dụng doanh nghiệp (EAI), cung cấp các tính năng kiểm soát quy trình đáng tin cậy và thực hiện các định dạng, giao thức dữ liệu internet. Hơn nữa, Python giúp người dùng xử lý các ngôn ngữ đánh dấu như XL, thực thi thông qua cùng một mã byte trên các hệ điều hành nâng cao và có thể được sử dụng như một ngôn ngữ kịch bản.

**Nhược điểm Python**

Cùng với một số ưu điểm, Python có một số hạn chế trong các lĩnh vực hiệu suất và bảo mật. Sau đây là một số nhược điểm đáng kể của việc sử dụng Python.

**Tốc độ thực thi chậm**

Python là một ngôn ngữ thông dịch, có nghĩa là nó hoạt động với trình thông dịch, không phải với trình biên dịch. Do đó, nó thực thi tương đối chậm hơn C, C ++, Java và nhiều ngôn ngữ khác.

**Tiêu thụ bộ nhớ lớn**

Các cấu trúc của Python đòi hỏi nhiều không gian bộ nhớ hơn. Ngôn ngữ này không thích hợp để sử dụng cho sự phát triển trong điều kiện bộ nhớ hạn chế.

**Không thích hợp cho phát triển trò chơi và thiết bị di động**

Python chủ yếu được sử dụng trong phát triển máy tính để bàn và web phía máy chủ. Nó không được coi là lý tưởng để phát triển ứng dụng di động và phát triển trò chơi do tiêu tốn nhiều bộ nhớ hơn và tốc độ xử lý chậm so với các ngôn ngữ lập trình khác.

**Hạn chế của Nhà phát triển**

Một khi nhà phát triển đã quen với sự dễ dàng và đơn giản của ngôn ngữ này, họ sẽ khó sử dụng các ngôn ngữ khác.

**Phát hiện lỗi trong mã**

Vì Python được thực thi thông qua trình thông dịch thay vì trình biên dịch, nên không thể phát hiện lỗi trong quá trình biên dịch và điều đó không tốt cho các nhà phát triển.

**Quyền truy cập cơ sở dữ liệu**

Python được coi là không an toàn cao và có nguy cơ bảo mật. Có một số hạn chế khi sử dụng Python để truy cập cơ sở dữ liệu. So với các công nghệ phổ biến khác như JDBC và ODBC, lớp truy cập cơ sở dữ liệu Python hơi kém phát triển và sơ khai.

**Hạn chế thiết kế**

Một trong những vấn đề quan trọng của Python là các hạn chế về thiết kế của nó.

**Khó kiểm tra**

Vì nó là một ngôn ngữ dựa trên trình thông dịch, rất khó để chạy các bài kiểm tra trên mã được viết bằng Python. Tất cả các lỗi chỉ xuất hiện trong thời gian chạy, điều này khiến việc kiểm tra các đoạn mã được viết bằng Python rất khó khăn

# **CHƯƠNG II: XÂY DỰNG CHƯƠNG TRÌNH**

# **2.1 Mô tả bài toán**

Hệ thống nhận diện đối tượng và trích xuất thông tin từ ảnh sẽ được xây dựng trên nền tảng ngôn ngữ lập trình Python, một ngôn ngữ mạnh mẽ và phổ biến trong việc xử lý ảnh và phát triển các mô hình học máy. Mô hình nhận diện khuôn mặt (Face Recognition Model) sẽ được sử dụng để nhận dạng và trích xuất thông tin từ các đối tượng trong hình ảnh. Hệ thống này sẽ được phát triển và triển khai trong môi trường lập trình Visual Studio Code (VSCode), kết hợp với các công cụ bổ trợ để hoàn thiện quy trình phát triển. Hình ảnh đầu vào sẽ được thu thập từ camera, từ đó hệ thống sử dụng mô hình nhận diện khuôn mặt để nhận dạng các đối tượng và trích xuất thông tin liên quan.

**2.3.1 Các điều kiện để xây dựng hệ thống**

Để xây dựng hệ thống nhận diện đối tượng và trích xuất thông tin từ ảnh, các điều kiện cần thiết bao gồm:

**Thu thập dữ liệu đối tượng**: Hệ thống cần thu thập dữ liệu hình ảnh từ các đối tượng cần nhận diện. Dữ liệu này có thể được thu thập thông qua việc chụp ảnh các đối tượng từ camera hoặc sử dụng các hình ảnh có sẵn. Mỗi đối tượng sẽ được chụp từ nhiều góc độ khác nhau để đảm bảo tính chính xác trong quá trình nhận diện.

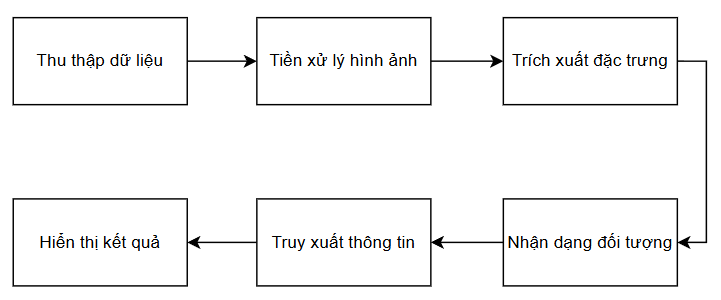
**Trích xuất đặc trưng của đối tượng và xuất file đặc trưng**: Sau khi thu thập hình ảnh, hệ thống sẽ tiến hành trích xuất các đặc trưng của đối tượng từ ảnh. Quá trình này sẽ sử dụng các thuật toán trích xuất đặc trưng như HOG, CNN, hoặc các phương pháp học sâu (deep learning) để thu được các mô tả đặc trưng của khuôn mặt hoặc đối tượng trong ảnh. Những đặc trưng này sẽ được mã hóa và lưu trữ vào các file dữ liệu, chẳng hạn như các file pickle, để chuẩn bị cho quá trình nhận dạng sau này.

**Nhận dạng đối tượng**: Mô hình học máy, đặc biệt là mô hình nhận diện khuôn mặt, sẽ được sử dụng để nhận dạng các đối tượng từ các đặc trưng đã được trích xuất. Mô hình này sẽ so sánh các đặc trưng đã được lưu trữ với các đặc trưng mới từ ảnh đầu vào để xác định đối tượng và phân loại chúng. Quá trình nhận dạng này sử dụng các kỹ thuật học sâu hoặc các phương pháp học máy khác, giúp hệ thống có thể nhận diện chính xác các đối tượng.

**Xuất thông tin đối tượng nhận dạng**: Sau khi nhận dạng thành công, hệ thống sẽ truy xuất và hiển thị thông tin liên quan đến đối tượng đã nhận dạng. Thông tin này có thể được lấy từ cơ sở dữ liệu, giúp người dùng dễ dàng nhận diện và xử lý các đối tượng trong hệ thống. Ví dụ, khi nhận diện khuôn mặt, hệ thống có thể hiển thị tên, địa chỉ hoặc các thông tin khác liên quan đến đối tượng đó.

# **2.2 Xây dựng hệ thống**

Chương trình nhận dạng đối tượng và trích xuất thông tin từ ảnh được chia thành nhiều giai đoạn, mỗi giai đoạn đảm nhiệm một vai trò cụ thể trong quy trình xử lý dữ liệu hình ảnh và nhận dạng khuôn mặt trong ảnh. Dưới đây là giải thích hình mô tả chi tiết về từng giai đoạn:



Hình 2.1 Sơ đồ tổng quát nhận dạng đối tượng trong ảnh

## **2.2.1 Thu Thập dữ liệu**

Giai đoạn thu thập dữ liệu là bước đầu tiên và quan trọng nhất trong quá trình xây dựng một hệ thống nhận dạng khuôn mặt. Đây là bước đặt nền tảng để đảm bảo chất lượng và hiệu quả của các bước tiếp theo như xử lý, nhận dạng và phân tích dữ liệu. Trong giai đoạn này, mục tiêu là thu thập được dữ liệu hình ảnh khuôn mặt có chất lượng cao, đa dạng, và phù hợp để huấn luyện và kiểm thử hệ thống.

**Vai trò và tầm quan trọng của thu thập dữ liệu**

Thu thập dữ liệu là bước thiết yếu vì chất lượng dữ liệu ảnh hưởng trực tiếp đến hiệu quả và độ chính xác của hệ thống nhận dạng khuôn mặt. Một tập dữ liệu tốt cần đáp ứng các tiêu chí sau:

* **Độ chính xác cao**: Hình ảnh khuôn mặt phải rõ ràng, chi tiết, và có thể nhận diện được các đặc điểm quan trọng như mắt, mũi, miệng, và hình dáng khuôn mặt.
* **Đa dạng**: Dữ liệu phải bao gồm nhiều yếu tố khác nhau như góc chụp, biểu cảm khuôn mặt, điều kiện ánh sáng, và nền môi trường để hệ thống hoạt động ổn định trong các điều kiện thực tế.
* **Tính đại diện**: Tập dữ liệu cần có tính đại diện cao, bao quát được các nhóm người dùng mục tiêu của hệ thống.

**Các nguồn thu thập dữ liệu**

Dữ liệu khuôn mặt có thể được thu thập từ nhiều nguồn khác nhau tùy thuộc vào ứng dụng cụ thể của hệ thống, bao gồm:

* **Chụp trực tiếp qua camera**: Đây là phương pháp phổ biến nhất, sử dụng camera để chụp ảnh hoặc quay video khuôn mặt. Phương pháp này đảm bảo dữ liệu thu thập được là chính xác và cập nhật.
* **Kho dữ liệu hình ảnh sẵn có**: Sử dụng các tập dữ liệu công khai như LFW (Labeled Faces in the Wild), CASIA-WebFace, hoặc các cơ sở dữ liệu thương mại để làm phong phú thêm tập dữ liệu.
* **Hình ảnh từ mạng xã hội hoặc internet**: Một số hệ thống có thể thu thập dữ liệu từ các nền tảng mạng xã hội hoặc tìm kiếm trực tuyến, nhưng cần chú ý đến các vấn đề pháp lý và quyền riêng tư.

**Quy trình thu thập dữ liệu**

Giai đoạn thu thập dữ liệu thường bao gồm các bước sau:

**Lựa chọn thiết bị và công cụ**

* Sử dụng camera có độ phân giải cao để đảm bảo hình ảnh rõ nét.
* Sử dụng các công cụ phát hiện khuôn mặt như OpenCV, Mediapipe, hoặc Haar Cascade để hỗ trợ việc chụp ảnh.

**Chuẩn bị môi trường chụp ảnh**

* **Ánh sáng**: Đảm bảo điều kiện ánh sáng đồng đều, tránh bóng tối hoặc ánh sáng quá mạnh gây khó khăn cho việc nhận dạng.
* **Phông nền**: Sử dụng phông nền trung tính để giảm nhiễu từ môi trường xung quanh.

**Thu thập dữ liệu từ nhiều góc độ và điều kiện**

* Chụp hình ảnh khuôn mặt từ các góc độ khác nhau (chính diện, góc nghiêng, từ trên xuống, từ dưới lên).
* Đa dạng hóa dữ liệu với các biểu cảm khác nhau (mỉm cười, nhăn mặt, không biểu cảm, v.v.).

**Lưu trữ dữ liệu một cách có tổ chức**

* Gán nhãn cho từng hình ảnh theo tên hoặc ID của đối tượng.
* Sắp xếp dữ liệu thành các thư mục hoặc bảng dữ liệu trong cơ sở dữ liệu.

**Các thách thức trong việc thu thập dữ liệu**

Việc thu thập dữ liệu khuôn mặt không phải lúc nào cũng dễ dàng và có thể gặp một số khó khăn như:

* **Chất lượng hình ảnh thấp**: Hình ảnh bị mờ, nhiễu, hoặc không đủ ánh sáng sẽ ảnh hưởng đến quá trình nhận dạng.
* **Tính riêng tư và pháp lý**: Thu thập hình ảnh khuôn mặt cần tuân thủ các quy định pháp luật liên quan đến quyền riêng tư và bảo mật dữ liệu.
* **Dữ liệu không đồng đều**: Nếu dữ liệu không đa dạng hoặc thiếu cân bằng, hệ thống có thể bị thiên vị đối với một nhóm người dùng nhất định.

## **2.2.2 Tiền xử lý hình ảnh**

Giai đoạn tiền xử lý hình ảnh là một bước quan trọng trong hệ thống nhận dạng khuôn mặt, nhằm cải thiện chất lượng hình ảnh và chuẩn bị dữ liệu để tối ưu hóa quá trình xử lý và phân tích. Mục tiêu của giai đoạn này là làm sạch dữ liệu hình ảnh, loại bỏ nhiễu, và chuẩn hóa các đặc điểm để hệ thống có thể nhận dạng khuôn mặt với độ chính xác cao hơn.

**Mục tiêu của giai đoạn tiền xử lý hình ảnh**

Giai đoạn tiền xử lý có vai trò quan trọng để:

* **Loại bỏ nhiễu**: Xử lý các yếu tố không mong muốn như ánh sáng yếu, độ mờ, hoặc nhiễu từ môi trường.
* **Chuẩn hóa hình ảnh**: Đảm bảo rằng tất cả các hình ảnh có cùng định dạng, kích thước, và đặc điểm đầu vào để thuận lợi cho các bước xử lý tiếp theo.
* **Tăng độ tương phản**: Làm rõ các đặc điểm quan trọng của khuôn mặt như mắt, mũi, miệng.
* **Giảm tải tính toán**: Tối ưu hóa kích thước và nội dung hình ảnh, giúp giảm thời gian xử lý và yêu cầu tài nguyên.

**Các bước trong giai đoạn tiền xử lý hình ảnh**

**Chuyển đổi hình ảnh sang mức xám (Grayscale Conversion)**: Hầu hết các hệ thống nhận dạng khuôn mặt không cần đến thông tin màu sắc mà chỉ tập trung vào đặc điểm hình dạng. Việc chuyển đổi hình ảnh sang mức xám giúp giảm số lượng kênh từ 3 (RGB) xuống 1, giảm tải tính toán và tăng hiệu quả xử lý.

**Cân bằng độ sáng và tương phản (Histogram Equalization)**: Cải thiện độ tương phản của hình ảnh bằng cách cân bằng biểu đồ tần số pixel. Phương pháp này giúp các đặc điểm khuôn mặt trở nên rõ ràng hơn trong các điều kiện ánh sáng không đồng đều.

**Làm mịn và loại bỏ nhiễu (Noise Reduction)** Sử dụng các bộ lọc như Gaussian Blur hoặc Median Filter để loại bỏ nhiễu trong hình ảnh và giảm thiểu các chi tiết không mong muốn mà vẫn giữ được các đường nét chính của khuôn mặt.

**Cắt và căn chỉnh khuôn mặt (Face Alignment and Cropping)**

Sử dụng các thuật toán phát hiện khuôn mặt như Haar Cascade, HOG + SVM, hoặc Mediapipe để xác định vị trí khuôn mặt trong hình ảnh.

**Chuẩn hóa kích thước (Resize)**: Đưa tất cả các hình ảnh về cùng một kích thước chuẩn (ví dụ: 128x128 pixel hoặc 256x256 pixel) để đảm bảo tính nhất quán trong việc xử lý dữ liệu đầu vào.

**Xử lý ánh sáng (Illumination Normalization)**: Giảm thiểu tác động của ánh sáng không đồng đều trên khuôn mặt bằng các phương pháp như Logarithmic Transformation hoặc Retinex Algorithm.

**Tách vùng quan trọng (Region of Interest - ROI)**: Tập trung vào các vùng quan trọng của khuôn mặt như mắt, mũi, miệng, bỏ qua các vùng không cần thiết như phông nền hoặc tóc.

## **2.2.3. Trích xuất đặc trưng**

Giai đoạn trích xuất đặc trưng là một bước quan trọng trong hệ thống nhận dạng khuôn mặt, giúp chuyển đổi hình ảnh đã được tiền xử lý thành các đặc điểm có thể sử dụng để nhận diện khuôn mặt. Mục tiêu chính của giai đoạn này là lấy ra các đặc điểm quan trọng từ khuôn mặt, giúp phân biệt các khuôn mặt với nhau một cách chính xác và hiệu quả.

**Mục tiêu của giai đoạn trích xuất đặc trưng**

Giai đoạn này nhằm đạt được các mục tiêu chính sau:

* **Chuyển đổi dữ liệu hình ảnh thành đặc trưng số**: Hình ảnh khuôn mặt được chuyển thành một tập hợp các đặc trưng (features) có thể dễ dàng phân tích và xử lý bởi các thuật toán học máy.
* **Tăng khả năng phân biệt**: Các đặc trưng được trích xuất giúp mô hình phân biệt rõ ràng giữa các khuôn mặt khác nhau.
* **Giảm chiều dữ liệu**: Thông qua việc trích xuất các đặc trưng quan trọng, ta có thể giảm số lượng thông tin cần xử lý mà không làm mất đi thông tin quan trọng.
* **Chuẩn hóa đặc trưng**: Các đặc trưng cần được chuẩn hóa để chúng có thể dễ dàng so sánh và sử dụng trong quá trình nhận dạng.

**Các phương pháp trích xuất đặc trưng phổ biến**

**Phương pháp trích xuất đặc trưng dựa trên hình ảnh (Traditional Methods)**

* **Haar-like Features (Haar Cascades)** :Đây là phương pháp sử dụng các đặc trưng tương tự như các bộ lọc Haar (một dạng đặc trưng hình học), phát hiện các khuôn mặt dựa trên sự thay đổi của mức độ sáng trong các vùng ảnh. Các đặc trưng này được sử dụng rộng rãi trong các ứng dụng nhận dạng khuôn mặt nhờ vào hiệu quả tính toán cao.
* **Local Binary Pattern (LBP)**: LBP là một phương pháp đơn giản nhưng hiệu quả để trích xuất các đặc trưng hình ảnh, đặc biệt trong nhận dạng khuôn mặt. LBP so sánh các điểm pixel với các điểm lân cận của chúng, tạo ra các mã nhị phân (binary code) cho mỗi điểm ảnh. Các mô hình LBP thường được sử dụng để phát hiện các đặc trưng hình học của khuôn mặt, chẳng hạn như mắt và mũi.
* **Scale-Invariant Feature Transform (SIFT) và Speeded-Up Robust Features (SURF):** Đây là các phương pháp phát hiện và mô tả các đặc trưng quan trọng trong hình ảnh, bao gồm các điểm đặc biệt (key points) và các vùng đặc trưng mạnh mẽ với sự thay đổi về góc nhìn, kích thước, và ánh sáng. Chúng rất hữu ích trong việc nhận dạng khuôn mặt ở nhiều góc độ và ánh sáng khác nhau.
* **Histogram of Oriented Gradients (HOG):** HOG là một phương pháp phân tích sự thay đổi độ sáng trong các vùng hình ảnh nhỏ để xác định các đặc trưng của đối tượng. Nó đặc biệt hiệu quả trong việc nhận diện các đặc điểm của khuôn mặt như mắt, mũi, miệng nhờ vào tính toán độ dốc của các hướng trong khu vực nhất định.

**Phương pháp trích xuất đặc trưng dựa trên học sâu (Deep Learning-Based Methods)**

* **Mạng nơ-ron tích chập (CNNs - Convolutional Neural Networks):** CNN là một trong những phương pháp mạnh mẽ nhất để trích xuất đặc trưng hình ảnh. Bằng cách sử dụng các lớp tích chập (convolution layers), các mạng CNN có thể học được các đặc trưng không gian phức tạp từ hình ảnh mà không cần sự can thiệp thủ công. Các đặc trưng này có thể bao gồm các đặc điểm đơn giản như cạnh và kết cấu, cho đến các đặc điểm phức tạp hơn như hình dạng khuôn mặt, mắt, và các đặc điểm khác.
* **FaceNet và DeepFace**: Đây là các mạng học sâu được thiết kế đặc biệt cho việc nhận dạng khuôn mặt. FaceNet, ví dụ, sử dụng mô hình mạng học sâu để trích xuất các đặc trưng hình ảnh và chuyển chúng thành các vector đặc trưng (embeddings). Các vector này có thể so sánh với nhau để xác định mức độ tương đồng giữa các khuôn mặt.
* **Deep Convolutional Neural Networks (DCNNs)**: Các mạng DCNNs như VGG-Face hoặc ResNet đã được chứng minh có hiệu quả cao trong việc trích xuất đặc trưng khuôn mặt. Những mạng này thường được huấn luyện với hàng triệu hình ảnh để học được các đặc trưng mạnh mẽ cho việc nhận dạng khuôn mặt.

## **2.2.4 Nhận dạng đối tượng**

Giai đoạn này được tiến hành sau khi quá trình trích xuất đặc trưng đã hoàn tất. Trong giai đoạn này, hệ thống sẽ thực hiện việc nhận diện các đối tượng có trong ảnh. Cụ thể, các đối tượng sẽ được phân tích và xác định, sau đó hệ thống sẽ tiến hành so sánh chúng với các đối tượng đã được trích xuất từ dữ liệu trước đó. Qua quá trình so sánh này, hệ thống sẽ xác định và nhận dạng chính xác các đối tượng trong ảnh, từ đó tạo ra kết quả nhận diện đáng tin cậy. Giai đoạn này đóng vai trò quan trọng trong việc đảm bảo rằng các thông tin và đặc trưng nhận diện được sử dụng một cách hiệu quả, giúp nâng cao độ chính xác của toàn bộ hệ thống.

**Mục tiêu của giai đoạn nhận diện đối tượng trong ảnh:**

**Xác định Đối Tượng**: Nhận diện và phân loại các đối tượng có trong ảnh một cách chính xác.

**So Sánh và Nhận Dạng**: So sánh các đối tượng đã nhận diện với dữ liệu đã trích xuất để xác định danh tính cụ thể của chúng.

**Tối Ưu Hóa Độ Chính Xác**: Nâng cao độ chính xác của quá trình nhận diện thông qua các thuật toán và kỹ thuật tiên tiến.

**Tăng Tốc Độ Xử Lý**: Cải thiện tốc độ xử lý để có thể nhận diện đối tượng trong thời gian thực, phục vụ cho các ứng dụng như giám sát an ninh, nhận diện khuôn mặt, v.v.

**Các Phương Pháp**

Để đạt được những mục tiêu trên, có thể áp dụng một số phương pháp sau:

**Học Sâu (Deep Learning):**

* Sử dụng các mạng nơ-ron tích chập (CNN) để tự động trích xuất và học các đặc trưng từ ảnh.
* Các mô hình như YOLO (You Only Look Once) hoặc Faster R-CNN có thể được sử dụng cho việc nhận diện đối tượng.

**Phân Tích Hình Ảnh (Image Processing):** Áp dụng các kỹ thuật xử lý hình ảnh như phát hiện biên, lọc và phân đoạn để cải thiện chất lượng ảnh trước khi nhận diện.

**Phương Pháp Học Máy (Machine Learning):** Sử dụng các thuật toán học máy truyền thống như SVM (Support Vector Machine) hoặc Decision Trees để phân loại đối tượng dựa vào các đặc trưng đã trích xuất.

**Kỹ Thuật Nhận Diện Khuôn Mặt (Face Recognition Techniques):** Sử dụng các mô hình chuyên biệt cho nhận diện khuôn mặt, như Eigenfaces hoặc Fisherfaces, để xác định danh tính trong các ứng dụng an ninh.

## **2.2.5. Truy xuất thông tin**

Tại giai đoạn trích xuất đặc trưng trong hệ thống nhận dạng khuôn mặt, sau khi quá trình nhận diện đối tượng (khuôn mặt) trong ảnh được thực hiện, một bước quan trọng tiếp theo là kiểm tra xem nhãn đối tượng đã được nhận diện (ví dụ, tên của người trong ảnh) có tồn tại trong cơ sở dữ liệu (CSDL) hay không. Nếu đối tượng đã tồn tại trong cơ sở dữ liệu, hệ thống sẽ tiến hành các bước tiếp theo để trích xuất thông tin liên quan đến đối tượng đó từ cơ sở dữ liệu. Quá trình này có thể được mô tả chi tiết như sau:

**Nhận diện đối tượng trong ảnh**

Sau khi hình ảnh được tiền xử lý và các đặc trưng khuôn mặt đã được trích xuất thành công, hệ thống nhận dạng sẽ so sánh các đặc trưng này với các đặc trưng đã được lưu trữ trong cơ sở dữ liệu để xác định đối tượng. Quá trình nhận diện này có thể sử dụng các mô hình học máy hoặc học sâu đã được huấn luyện trước, chẳng hạn như các mô hình CNN (Mạng Nơ-ron Tích chập) hay các mô hình FaceNet.

Khi hệ thống xác định được đối tượng trong ảnh (ví dụ, một khuôn mặt cụ thể), nhãn đối tượng đó sẽ được gán, thường là tên hoặc ID của người có trong ảnh. Sau đó, hệ thống sẽ chuyển sang bước kiểm tra thông tin về đối tượng này trong cơ sở dữ liệu.

**Kiểm tra sự tồn tại của nhãn đối tượng trong cơ sở dữ liệu**

Bước tiếp theo sau khi nhận diện được nhãn đối tượng là kiểm tra xem đối tượng này có tồn tại trong cơ sở dữ liệu hay không. Cơ sở dữ liệu có thể chứa các thông tin liên quan đến nhiều đối tượng đã được ghi nhận từ trước, chẳng hạn như tên, ảnh, thông tin cá nhân, lịch sử giao dịch hoặc các dữ liệu liên quan khác.

Hệ thống sẽ thực hiện một truy vấn tìm kiếm trong cơ sở dữ liệu dựa trên nhãn đối tượng (ví dụ: tên của người được nhận diện hoặc mã số nhận dạng khuôn mặt). Nếu nhãn này tồn tại trong cơ sở dữ liệu, hệ thống sẽ xác nhận sự tồn tại của đối tượng và chuyển sang bước tiếp theo.

**Trích xuất thông tin từ cơ sở dữ liệu**

Nếu đối tượng được nhận diện và nhãn của nó tồn tại trong cơ sở dữ liệu, hệ thống sẽ tiến hành trích xuất các thông tin liên quan đến đối tượng đó từ cơ sở dữ liệu. Các thông tin này có thể bao gồm: Thông tin cá nhân: Như tên, tuổi, giới tính, địa chỉ, số điện thoại, hoặc bất kỳ thông tin nào mà cơ sở dữ liệu lưu trữ về đối tượng.

Việc trích xuất các thông tin này sẽ giúp hệ thống cung cấp thông tin chi tiết và chính xác về đối tượng cho người dùng hoặc hệ thống để tiếp tục thực hiện các tác vụ liên quan, chẳng hạn như xác minh danh tính, cung cấp dịch vụ hoặc tiến hành các hành động khác dựa trên thông tin đã có.

**Nếu đối tượng không tồn tại trong cơ sở dữ liệu**

Trong trường hợp đối tượng không tồn tại trong cơ sở dữ liệu, hệ thống có thể thực hiện một số hành động khác như:

* Cảnh báo cho người dùng hoặc hệ thống: Nếu không tìm thấy đối tượng trong cơ sở dữ liệu, hệ thống có thể thông báo cho người dùng hoặc yêu cầu hành động khác, chẳng hạn như thêm thông tin mới vào cơ sở dữ liệu.
* Cập nhật hoặc thêm mới vào cơ sở dữ liệu: Hệ thống có thể yêu cầu cập nhật hoặc thêm mới thông tin của đối tượng vào cơ sở dữ liệu, để lần sau có thể nhận diện và lưu trữ thông tin của đối tượng một cách chính xác.

## **2.2.6. Mô hình nhận diện khuôn mặt (Face Recognition Model)**

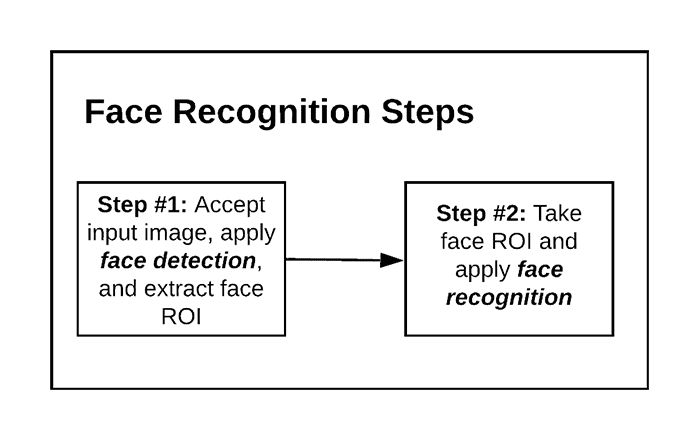
Face recognition (nhận diện khuôn mặt) là quá trình lấy một bức ảnh chứa mặt người và xác định ai trong đó.

Ban đầu hệ thống nhận diện khuôn mặt dựa trên facial landmarks từ ảnh như vị trí, kích thước của mắt, mũi, miệng, má. Tuy nhiên độ chính xác của phương pháp này không cao. Sau này các kỹ thuật toán ML phát triển các đặc trưng được tách ra từ khuôn mặt, các model được huấn luyện để phân loại từ đó xác định khuôn mặt tương ứng với ai. Gần đây nhất các kỹ thuật Deep Learning được sử dụng để nhận dạng khuôn mặt như model FaceNet, OpenFace dựa trên kiến trúc Siamese Network cho kết quả rất tốt.

**Nhận diện khuôn mặt khác phát hiện khuôn mặt**

Phát hiện khuôn mặt là xác định vị trí của khuôn mặt trong hình ảnh, video. Nhận diện khuôn mặt là vừa xác định vị trí khuôn mặt vừa xác định đó là ai. Bài toàn này phức tạp hơn và bao trùm phát hiện khuôn mặt (face detection). Quá trình nhận diện khuôn mặt bao gồm 2 bước:

* Face detection: sử dụng các phương pháp như Harr cascades, HOG + Linear SVM, deep learning (MTCNN…)
* Lấy khuôn mặt phát hiện được rồi xác định ai trong đó: ví dụ sử dụng model FaceNet dựa trên Siamese Network, Local Bianry Patterns (LBPs), Eigenfaces…



Hình 2.1.2 Các bước trong nhận diện đối tượng

Thực chất có thể gộp 2 bước làm một tuy nhiên làm như vậy ảnh sẽ có nhiều nhiễu (background) ảnh hưởng đến độ chính xác.

**Ưu điểm:**

**Độ chính xác cao:CNN (Convolutional Neural Networks)** có khả năng nhận diện khuôn mặt với độ chính xác rất cao, đặc biệt khi có đủ dữ liệu huấn luyện. CNN có thể nhận diện khuôn mặt ngay cả khi có thay đổi về góc độ, ánh sáng, hoặc biểu cảm khuôn mặt.

**Dễ triển khai:** Thư viện **face\_recognition** dễ sử dụng và không yêu cầu quá nhiều kiến thức chuyên sâu về học máy. Việc cài đặt và triển khai mô hình nhận diện khuôn mặt khá đơn giản.

**Hiệu suất tốt đối với ảnh tĩnh:** Mô hình này hoạt động hiệu quả với ảnh tĩnh, khi khuôn mặt có thể được phát hiện và mã hóa chính xác. Các ảnh chất lượng tốt với ánh sáng đầy đủ sẽ cho kết quả tốt nhất.

**Tích hợp với Firebase:** Mô hình có khả năng truy vấn dữ liệu người dùng từ Firebase, cho phép dễ dàng lưu trữ và truy xuất thông tin về các khuôn mặt đã nhận diện.

**Phát hiện và nhận diện nhiều khuôn mặt:**Mô hình có thể xử lý nhiều khuôn mặt trong một ảnh, cho phép nhận diện và phân biệt giữa nhiều người trong một cảnh.

**Nhược điểm:**

**Yêu cầu về phần cứng mạnh mẽ:** **CNN** có thể yêu cầu phần cứng mạnh mẽ (như GPU) để đạt hiệu suất tối ưu, đặc biệt trong trường hợp xử lý nhiều ảnh hoặc video thời gian thực. Việc xử lý ảnh với phương pháp **CNN** có thể chậm hơn so với **HOG**.

**Khả năng nhận diện hạn chế trong điều kiện ánh sáng kém:** Mặc dù mô hình này có thể hoạt động trong nhiều điều kiện ánh sáng khác nhau, nhưng độ chính xác của nhận diện sẽ giảm khi môi trường có ánh sáng yếu hoặc khuôn mặt bị che khuất.

**Phải có đủ dữ liệu huấn luyện cho mỗi người:** Mô hình yêu cầu dataset có đủ ảnh khuôn mặt của mỗi người để tạo ra các encoding chính xác. Nếu không đủ ảnh mẫu hoặc ảnh không đủ chất lượng, kết quả nhận diện có thể không chính xác.

**Chậm với số lượng ảnh lớn:** Khi xử lý một số lượng lớn ảnh hoặc video thời gian thực, mô hình có thể gặp khó khăn về tốc độ, đặc biệt khi sử dụng **CNN** thay vì **HOG**. **HOG** có thể nhanh hơn nhưng ít chính xác hơn so với **CNN**.

**Không hoàn hảo với ảnh mờ hoặc có nhiễu:**Nếu ảnh có độ phân giải thấp, mờ, hoặc có nhiều nhiễu, khả năng nhận diện khuôn mặt sẽ bị ảnh hưởng đáng kể, dẫn đến các lỗi nhận diện hoặc nhận diện sai.

**Nhạy cảm với góc nhìn và biểu cảm khuôn mặt:**Mặc dù mô hình khá mạnh mẽ, nhưng nó vẫn có thể gặp khó khăn trong việc nhận diện khi khuôn mặt bị che khuất, quay ngang hoặc nghiêng, hoặc khi người đó có biểu cảm khuôn mặt thay đổi mạnh (ví dụ, cười, khóc).

Qua đó cho ta thấy được rằng, mô hình nhận diện khuôn mặt của bạn rất hữu ích trong các tình huống với dữ liệu ảnh chất lượng cao và điều kiện ánh sáng ổn định. Tuy nhiên, để sử dụng hiệu quả trong môi trường thời gian thực hoặc với số lượng lớn ảnh, bạn có thể cần tối ưu hóa về phần cứng hoặc cân nhắc sử dụng các phương pháp khác để cải thiện tốc độ và độ chính xác.

## **2.2.7 Thuật toán CNN**

Mạng nơ-ron tích chập (Convolutional Neural Network - CNN) là một loại mạng nơ-ron sâu (Deep Neural Network) được thiết kế đặc biệt để xử lý dữ liệu có cấu trúc dạng lưới, chẳng hạn như hình ảnh. CNN đã trở thành một trong những phương pháp phổ biến nhất trong lĩnh vực nhận diện hình ảnh và thị giác máy tính.

**Cấu trúc của CNN**

**Một mạng CNN thường bao gồm các thành phần chính sau:**

**Lớp Tích chập (Convolutional Layer):**

* Đây là lớp chính của CNN, nơi mà các bộ lọc (filters) được áp dụng để trích xuất các đặc trưng từ hình ảnh. Mỗi bộ lọc học được các đặc trưng khác nhau như đường viền, góc cạnh, và hình dạng.
* Kết quả của lớp này là một ma trận các đặc trưng (feature maps).

**Lớp Kích hoạt (Activation Layer):**

* Sau mỗi lớp tích chập, một hàm kích hoạt, thường là ReLU (Rectified Linear Unit), được áp dụng để giới thiệu tính phi tuyến vào mô hình.

**Lớp Gộp (Pooling Layer):**

* Lớp này giảm kích thước của ma trận đặc trưng, giúp giảm số lượng tham số và tính toán, đồng thời giảm thiểu hiện tượng quá khớp (overfitting).
* Các phương pháp gộp phổ biến là max pooling và average pooling.

**Lớp Fully Connected (FC Layer):**

* Cuối cùng, các đặc trưng được đưa vào một hoặc nhiều lớp fully connected để đưa ra dự đoán cuối cùng. Mỗi nơ-ron trong lớp này kết nối với tất cả các nơ-ron trong lớp trước.

**Ưu điểm của CNN**

* **Tự động trích xuất đặc trưng**: CNN tự động học và trích xuất các đặc trưng quan trọng từ dữ liệu mà không cần phải làm thủ công.
* **Giảm số lượng tham số**: Nhờ vào việc sử dụng các bộ lọc nhỏ và pooling, CNN có thể giảm thiểu số lượng tham số so với các mạng fully connected truyền thống.
* **Khả năng tổng quát**: CNN có khả năng tổng quát tốt hơn cho các dữ liệu chưa thấy, đặc biệt trong việc nhận diện các đối tượng trong hình ảnh.

# **2.2 Các giai đoạn hệ thông**

## **2.2.1. Giai đoạn quản lý dữ liệu**

Quản lý dữ liệu là một bước quan trọng trong việc xây dựng mô hình học máy để nhận diện đối tượng và trích xuất thông tin từ ảnh chụp bằng camera. Giai đoạn này bao gồm thu thập, tổ chức và chuẩn bị dữ liệu từ các ảnh chụp thực tế để có thể sử dụng hiệu quả trong quá trình huấn luyện và nhận diện đối tượng.

**Thu thập Dữ liệu:**

* Dữ liệu đầu vào là các hình ảnh được chụp từ camera, có thể là camera của điện thoại di động, máy tính hoặc camera chuyên dụng. Những hình ảnh này chứa các đối tượng mà mô hình cần nhận diện.
* Các ảnh thu thập từ camera có thể được ghi lại từ các tình huống thực tế như quét mã vạch, nhận diện khuôn mặt, phân loại vật thể trong môi trường, hoặc nhận diện chữ viết tay trong hình ảnh.
* Dữ liệu thu thập sẽ được tổ chức trong các thư mục riêng biệt, mỗi thư mục chứa các hình ảnh của một đối tượng cụ thể (ví dụ: các vật thể hoặc các loại chữ viết tay). Mỗi hình ảnh sẽ được gắn nhãn với thông tin chi tiết về đối tượng mà nó đại diện (ví dụ: tên đối tượng, vị trí trong ảnh, hoặc các thuộc tính đặc biệt của đối tượng).

## **2.2.2. Giai đoạn huận luyện mô hình học máy**

Huấn luyện mô hình là bước quan trọng trong việc phát triển hệ thống nhận diện đối tượng và trích xuất thông tin từ ảnh. Giai đoạn này liên quan đến việc mô hình học máy học từ các dữ liệu đã chuẩn bị, trích xuất đặc trưng từ ảnh và lưu trữ chúng vào file để tiến hành nhận dạng.

**Chuẩn Bị Dữ liệu Huấn luyện:**

* Dữ liệu huấn luyện trong hệ thống nhận diện khuôn mặt được thu thập từ các hình ảnh chụp từ camera. Các hình ảnh này có thể là ảnh chụp của các khuôn mặt từ nhiều góc độ và điều kiện ánh sáng khác nhau.
* Dữ liệu huấn luyện sẽ được chia thành hai phần: dữ liệu đặc trưng (các đặc điểm khuôn mặt được trích xuất từ ảnh) và nhãn (danh tính hoặc tên của người trong ảnh).
* Dữ liệu huấn luyện cần được chia thành hai tập: một tập huấn luyện (train) và một tập kiểm tra (test). Tập huấn luyện sẽ được sử dụng để huấn luyện mô hình nhận diện khuôn mặt, trong khi tập kiểm tra giúp đánh giá độ chính xác của mô hình.

**Chọn Mô hình Học Máy:**

* Trong bài toán này, mô hình học máy được lựa chọn là mô hình nhận diện khuôn mặt (Face Recognition Model). Mô hình này sẽ sử dụng các thuật toán học máy để nhận diện các đặc điểm khuôn mặt và phân biệt chúng.
* Một số mô hình phổ biến cho nhận diện khuôn mặt bao gồm các phương pháp dựa trên Deep Learning như mạng nơ-ron tích chập (CNN) hoặc các mô hình đặc biệt như FaceNet, OpenFace, hoặc mô hình học máy sử dụng các đặc trưng tĩnh của khuôn mặt.

**Quá Trình Huấn Luyện:**

* Quá trình huấn luyện sẽ bao gồm việc chuyển đổi dữ liệu hình ảnh từ camera thành một tập hợp các đặc trưng khuôn mặt mã hóa. Các đặc trưng này có thể là các vector tính toán từ các đặc điểm khuôn mặt, chẳng hạn như khoảng cách giữa các mắt, hình dáng của mũi, hoặc các điểm nổi bật trên khuôn mặt.
* Dữ liệu hình ảnh từ các khuôn mặt sẽ được chuyển đổi thành một file dữ liệu mã hóa (ví dụ: file CSV hoặc file nhị phân), trong đó mỗi dòng đại diện cho một khuôn mặt và chứa các đặc trưng của khuôn mặt đó.
* Sau khi dữ liệu được mã hóa, mô hình sẽ được huấn luyện để nhận diện các khuôn mặt và gán nhãn đúng cho từng khuôn mặt trong tập huấn luyện.

**Đánh Giá Mô hình:**

* Sau khi huấn luyện xong, mô hình sẽ được kiểm tra với tập dữ liệu kiểm tra. Mô hình sẽ đưa ra dự đoán cho các khuôn mặt trong tập kiểm tra và so sánh với nhãn thực tế (danh tính của người trong ảnh) để tính toán độ chính xác của mô hình.
* Nếu độ chính xác của mô hình đạt yêu cầu, mô hình có thể được lưu lại để sử dụng trong các bước nhận diện khuôn mặt trong các ứng dụng thực tế.

## **2.2.3. Giai đoạn nhận dạng đối tượng**

**Thu Thập Dữ Liệu Đầu Vào:**

* Dữ liệu đầu vào là các hình ảnh chụp từ camera hoặc ảnh có sẵn được sử dụng để nhận diện đối tượng. Các hình ảnh này có thể là ảnh của các đối tượng cụ thể hoặc khuôn mặt, tùy thuộc vào mục tiêu nhận diện.
* Dữ liệu đầu vào sẽ được chuyển đổi thành các **đặc trưng** trong quá trình tiền xử lý, và được lưu dưới dạng các file dữ liệu đã xuất đặc trưng (ví dụ: file CSV hoặc file chứa các giá trị đặc trưng của ảnh).

**Tiền Xử Lý Dữ Liệu Nhận Dạng:**

Dữ liệu đầu vào từ camera hoặc các hình ảnh đã có sẵn sẽ được trích xuất đặc trưng và lưu vào một file. Quá trình tiền xử lý này bao gồm các bước sau:

* **Trích xuất đặc trưng từ ảnh**: Các đặc trưng (như các điểm đặc trưng, vector đặc trưng, hoặc các đặc tính hình học khác) được trích xuất từ các hình ảnh đầu vào. Các đặc trưng này giúp mô hình nhận dạng có thể phân biệt đối tượng trong ảnh.
* **Lưu đặc trưng vào file dữ liệu**: Các đặc trưng được trích xuất sẽ được lưu vào file dữ liệu (ví dụ: một file CSV chứa các đặc trưng của ảnh). Mỗi dòng trong file sẽ chứa các đặc trưng (giá trị pixel, đặc trưng hình học, v.v.) của một đối tượng hoặc khuôn mặt trong ảnh.
* **Chuẩn bị dữ liệu cho mô hình nhận dạng**: Các file dữ liệu đặc trưng này sẽ được sử dụng làm đầu vào cho mô hình học máy trong giai đoạn nhận dạng.

**Dự Đoán Với Mô Hình:**

* Sau khi dữ liệu đặc trưng được trích xuất và lưu vào file, mô hình nhận dạng đối tượng (ví dụ: mô hình nhận dạng khuôn mặt hoặc mô hình nhận diện đối tượng khác) sẽ sử dụng các đặc trưng trong file để đưa ra dự đoán.
* Mô hình sẽ phân tích các đặc trưng từ file dữ liệu và đưa ra kết quả dự đoán về đối tượng hoặc danh tính của đối tượng trong ảnh.

**Hiển Thị Kết Quả:**

* Kết quả dự đoán sẽ được hiển thị cho người dùng dưới dạng nhãn mô tả đối tượng nhận diện, ví dụ: "Dự đoán: Người A" hoặc "Dự đoán: Mặt người".
* Hệ thống có thể hiển thị hình ảnh của đối tượng đã nhận diện cùng với kết quả dự đoán, giúp người dùng kiểm tra và xác nhận kết quả.

## **2.2.4. Giai đoạn trích xuất thông tin**

Sau khi hệ thống hoàn thành việc nhận dạng đối tượng từ hình ảnh đầu vào, nó sẽ kết nối tới cơ sở dữ liệu Firebase để tiến hành tìm kiếm thông tin liên quan đến đối tượng. Cụ thể, hệ thống sẽ sử dụng nhãn dán (label) của đối tượng đã nhận dạng được để thực hiện truy vấn vào Firebase. Nhãn dán này giúp xác định đối tượng cần tìm kiếm trong cơ sở dữ liệu, từ đó lấy được thông tin chi tiết về đối tượng đó. Quá trình này giúp hệ thống cung cấp các thông tin liên quan hoặc thực hiện các hành động tiếp theo, như hiển thị thông tin của đối tượng, cập nhật cơ sở dữ liệu, hoặc thực hiện các tác vụ khác theo yêu cầu của ứng dụng.

# **2.3 Xây dựng hệ thống**

Hệ thống nhận diện đối tượng và trích xuất thông tin từ ảnh sẽ được xây dựng trên nền tảng ngôn ngữ lập trình Python, một ngôn ngữ mạnh mẽ và phổ biến trong việc xử lý ảnh cũng như phát triển các mô hình học máy. Mô hình nhận diện khuôn mặt (Face Recognition Model) sẽ được sử dụng để nhận dạng và trích xuất thông tin từ các đối tượng trong hình ảnh. Hệ thống sẽ được phát triển và cài đặt trên môi trường lập trình Visual Studio Code (VSCode), kết hợp với các công cụ bổ trợ khác để hoàn thiện quy trình phát triển. Hình ảnh đầu vào sẽ được lấy từ camera, từ đó hệ thống sẽ sử dụng mô hình nhận diện khuôn mặt để nhận dạng các đối tượng và trích xuất thông tin liên quan.

## **2.3.1 Các điều kiện để xây dựng hệ thống**

**Để xây dựng hệ thống nhận diện đối tượng và trích xuất thông tin từ ảnh, các điều kiện cần thiết bao gồm:**

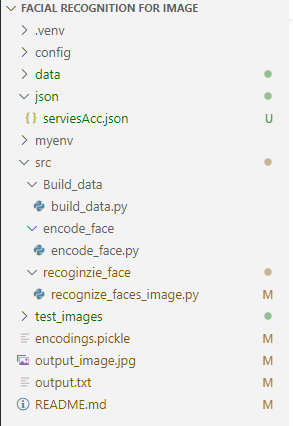
Thu thập dữ liệu đối tượng: Hệ thống cần thu thập dữ liệu hình ảnh từ các đối tượng cần nhận diện. Dữ liệu này có thể được thu thập thông qua việc chụp ảnh các đối tượng từ camera.

Trích xuất đặc trưng của đối tượng và xuất file đặc trưng: Sau khi thu thập hình ảnh, hệ thống sẽ tiến hành trích xuất các đặc trưng từ hình ảnh của đối tượng. Những đặc trưng này sẽ được mã hóa và lưu vào các file dữ liệu để chuẩn bị cho quá trình nhận dạng sau này.

Nhận dạng đối tượng: Mô hình học máy (cụ thể là mô hình nhận diện khuôn mặt) sẽ được sử dụng để nhận dạng các đối tượng từ các đặc trưng đã được trích xuất. Quá trình này giúp xác định các đối tượng trong ảnh và phân loại chúng.

Xuất thông tin đối tượng nhận dạng: Sau khi nhận dạng thành công, hệ thống sẽ truy xuất và hiển thị thông tin liên quan đến đối tượng đã nhận dạng. Thông tin này có thể được lấy từ cơ sở dữ liệu, giúp người dùng dễ dàng nhận diện và xử lý các đối tượng trong hệ thống.

## **2.3.2 Tổ chức và quản lý dự án**

****

**venv**: Đây là môi trường ảo cho dự án, nơi các thư viện (như các thư viện Python) được cài đặt để tránh xung đột với các dự án khác. Thư mục này thường bị ẩn mặc định.

**config**: Thư mục này có thể chứa các tệp cấu hình cho dự án, chẳng hạn như cài đặt cho các thư viện hoặc các tham số cho mô hình nhận diện khuôn mặt.

**data**: Thư mục này có thể chứa dữ liệu thô được sử dụng để huấn luyện hoặc kiểm tra hệ thống nhận diện khuôn mặt, ví dụ như hình ảnh hoặc các bộ dữ liệu khác.

**dlib**: Thư mục này có thể liên quan đến thư viện dlib hoặc các tệp liên quan. Dlib là một thư viện học máy phổ biến được sử dụng trong nhận diện khuôn mặt và phát hiện đối tượng.

**json**: Thư mục này có thể chứa các tệp JSON, dùng để lưu trữ dữ liệu theo định dạng cấu trúc, chẳng hạn như metadata, cấu hình hoặc thông tin về mô hình.

**myenv**: Đây có thể là một môi trường ảo khác hoặc là thư mục chứa cấu hình hoặc tập lệnh riêng cho người dùng.

**src**: Thư mục này chứa mã nguồn của dự án. Bên trong thư mục này có thể có các thư mục con cho các phần khác nhau của mã nguồn.

**Build\_data**: Thư mục này có thể chứa các tập lệnh để chuẩn bị hoặc tiền xử lý dữ liệu trước khi sử dụng trong mô hình.

**encode\_face:** Thư mục này có thể chứa các tập lệnh hoặc hàm để mã hóa đặc điểm khuôn mặt, chuyển khuôn mặt thành các đại diện số để sử dụng trong nhận diện.

**recognize\_face**: Thư mục này có thể chứa mã nguồn thực hiện thuật toán nhận diện khuôn mặt hoặc mô hình nhận diện, xử lý các khuôn mặt đã được mã hóa để xác định hoặc so khớp chúng.

# **CHƯƠNG III: THỰC NGHIỆM KẾT QUẢ**

# **3.1. Ngôn ngữ lập trình và thư viện sử dụng**

## **3.1.1. 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). Các nhà phát triển sử dụng Python vì nó hiệu quả, dễ học và có thể chạy trên nhiều nền tảng khác nhau. Phần mềm Python được tải xuống miễn phí, tích hợp tốt với tất cả các loại hệ thống và tăng tốc độ phát triển.

## **3.1.2 Những thư viện trong học máy được sử dụng trong bài toán.**

**3.1.2.1. OpenCV**

**Lý thuyết:** OpenCV (Open Source Computer Vision Library) là một thư viện mã nguồn mở chuyên về xử lý ảnh và thị giác máy tính. Nó cung cấp các công cụ để xử lý hình ảnh, nhận diện đối tượng, phát hiện khuôn mặt, và nhiều ứng dụng khác.

**Ưu điểm:**

* + Mã nguồn mở và miễn phí.
  + Được hỗ trợ rộng rãi với cộng đồng người dùng lớn.
  + Tích hợp nhiều công cụ xử lý ảnh mạnh mẽ, tối ưu hóa cho thời gian thực.
  + Hỗ trợ đa nền tảng (Windows, Linux, macOS, Android, iOS).

**Nhược điểm:**

* + Cấu trúc phức tạp, có thể khó học đối với người mới.
  + Không tích hợp sẵn các thuật toán học sâu; cần kết hợp với các thư viện khác như TensorFlow, PyTorch.

**3.1.2.2. os**

**Lý thuyết:** os là một thư viện tích hợp trong Python, cung cấp các chức năng để tương tác với hệ điều hành như quản lý tệp, thư mục, và thực hiện các lệnh hệ thống.

**Ưu điểm:**

* Đơn giản, dễ sử dụng.
* Được tích hợp sẵn, không cần cài đặt thêm.
* Cho phép tương tác trực tiếp với tệp và thư mục trên hệ thống.

**Nhược điểm:**

* + Không hỗ trợ các tác vụ phức tạp mà cần phải kết hợp với các thư viện khác.

**3.1.2.3. argparse**

**Lý thuyết:** argparse là một thư viện tích hợp trong Python dùng để phân tích các tham số dòng lệnh khi chạy chương trình.

**Ưu điểm:**

* Dễ sử dụng, cung cấp cú pháp rõ ràng.
* Hỗ trợ tự động tạo các thông báo lỗi và hướng dẫn sử dụng.
* Linh hoạt và mạnh mẽ khi làm việc với các tham số phức tạp.

**Nhược điểm:**

* Hạn chế trong việc xử lý nhiều luồng tác vụ hoặc giao diện người dùng phức tạp.

**3.1.2.4. imutils**

**Lý thuyết:** imutils là một thư viện tiện ích hỗ trợ xử lý hình ảnh với OpenCV. Nó đơn giản hóa các tác vụ thường gặp như xoay ảnh, thay đổi kích thước, hay xử lý tọa độ.

**Ưu điểm:**

* Tích hợp các chức năng thông dụng, giúp giảm mã lệnh.
* Dễ sử dụng, phù hợp với người mới.

**Nhược điểm:**

* Chỉ là thư viện bổ sung, không cung cấp các công cụ xử lý nâng cao.

**3.1.2.5. pickle**

**Lý thuyết:** pickle là một thư viện Python dùng để tuần tự hóa (serialize) và giải tuần tự hóa (deserialize) các đối tượng Python, giúp lưu trữ hoặc truyền dữ liệu dễ dàng.

**Ưu điểm:**

* Đơn giản, dễ sử dụng.
* Hỗ trợ hầu hết các kiểu dữ liệu Python.

**Nhược điểm:**

* Bảo mật thấp: có thể thực thi mã độc từ tệp pickle.
* Tốc độ không tối ưu với dữ liệu lớn hoặc phức tạp.

**3.1.2.6. face\_recognition**

**Lý thuyết:** face\_recognition là một thư viện Python được xây dựng trên dlib, hỗ trợ nhận diện và nhận dạng khuôn mặt một cách hiệu quả.

**Ưu điểm:**

* Cực kỳ dễ sử dụng với API đơn giản.
* Tích hợp sẵn nhiều công cụ mạnh mẽ cho nhận diện khuôn mặt.
* Chính xác và hiệu quả khi sử dụng trên dữ liệu khuôn mặt lớn.

**Nhược điểm:**

* Yêu cầu phần cứng mạnh khi xử lý nhiều hình ảnh hoặc video.
* Không được cập nhật thường xuyên.

**3.1.2.7. firebase\_admin**

**Lý thuyết:** firebase\_admin là thư viện Python hỗ trợ tương tác với Firebase, một nền tảng phát triển ứng dụng của Google. Nó giúp quản lý cơ sở dữ liệu thời gian thực, xác thực người dùng, và lưu trữ dữ liệu.

**Ưu điểm:**

* Kết nối dễ dàng với Firebase và các dịch vụ của Google.
* Cung cấp khả năng đồng bộ dữ liệu thời gian thực.
* Hỗ trợ mở rộng ứng dụng lớn với backend mạnh mẽ.

**Nhược điểm:**

* Phụ thuộc vào dịch vụ của Google.
* Cần kết nối internet để hoạt động.
* Không phải là mã nguồn mở.

**3.1.2.8. tabulate**

**Lý thuyết:** tabulate là một thư viện Python giúp trình bày dữ liệu dạng bảng trong giao diện dòng lệnh một cách dễ đọc.

**Ưu điểm:**

* Dễ sử dụng, phù hợp cho việc hiển thị dữ liệu đơn giản.
* Hỗ trợ nhiều định dạng bảng khác nhau.

**Nhược điểm:**

* Giới hạn trong việc xử lý dữ liệu phức tạp.
* Không hỗ trợ giao diện đồ họa.

# **3.2. Các bước chạy chương trình**

**Bước1. Tạo dataset**

Ở đây chúng ta sử dụng **build\_data.py** để xây dựng dataset. Trong thư mục src/Build\_data có chứa các subdirctory cho từng người với tên (+ ID nếu cần), trong mỗi subdirectory lại chứa ảnh khuôn mặt của người đó.

|  |  |
| --- | --- |
|  |  |

Hình 3.1.1 Hình ảnh đối tượng

Chú ý: Mỗi ảnh nên chỉ chứa duy nhất 1 khuôn mặt của người đó (nếu có các khuôn mặt của nhiều người, phần implementation bên dưới sẽ phức tạp hơn vì đôi khi phải xác định thêm ai trong bức ảnh)

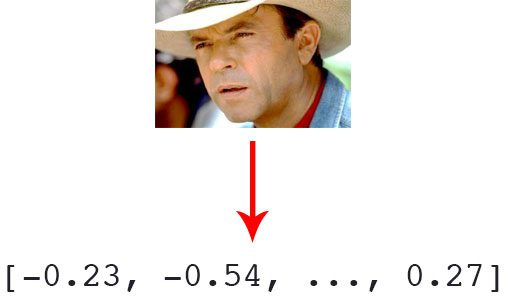
Ở đây dataset được tạo thông qua webcam. Đưa mặt người đến gần, xa webcam, với các tư thế, biểu cảm khác nhau. Chạy file **build\_dataset.py** rồi nhấn phím “k” để lưu ảnh của từng người. Đối với mỗi người nên có ít nhất 10-20 ảnh để mô hình có độ chính xác cao. Trong file này mình không thêm bộ phát hiện khuôn mặt vào ví dụ Haar cascades hoặc kẻ khung để người dùng điều chỉnh mặt trong đó. Ở đây mình muốn mô hình nhận được ảnh chứa khuôn mặt với nhiều điều kiện khác nhau để mô hình có thể hoạt động tốt hơn trong thực tế (lưu thêm nhiều ảnh cho nhiều điều kiện).

Ngoài việc tạo dataset thông qua webcam, chúng ta có thể tạo dataset thủ công hoặc sử dụng Search API như Bing hay Google.

Sau khi dataset được tạo với **build\_dataset.py** chúng ta sẽ chạy **encode\_faces.py** để tạo các embeddings.

**Bước 2. Tạo encodings cho các khuôn mặt trong dataset**

Sau khi tạo xong dataset chúng ta sẽ đi tạo các encodings (hay embeddings) của các khuôn mặt trong dataset đó. Việc đầu tiên cần làm là đi trích xuất các face ROIs (tránh sử dụng hết cả ảnh vì sẽ có nhiều nhiễu background ảnh hưởng đến chất lượng mô hình). Để phát hiện và trích xuất khuôn mặt có thể sử dụng nhiều phương pháp như haar cascades, HOG + Linear SVM, Deep Learning-bases face detector... Khi có các face ROIs chúng ta sẽ đưa chúng qua mạng NN để lấy các encodings.

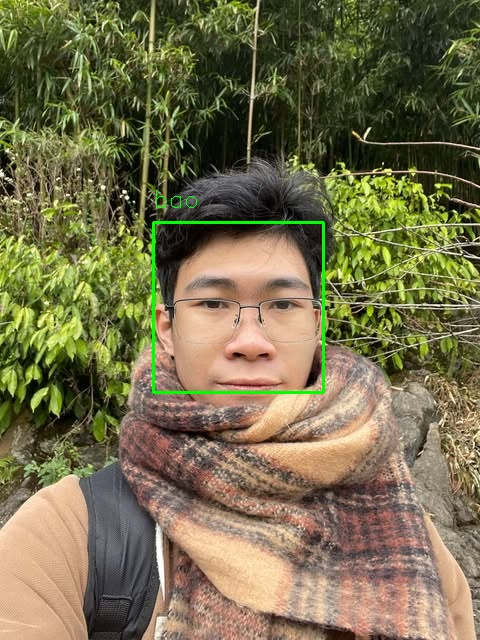


Hình 3.1.2: HìnhTạo encoding từ ảnh khuôn mặt

Ở đây chúng ta không training lại từ đầu mạng tạo encodings mà sử dụng lại pre-trained model (trong thư viện **dlib** và được tích hợp vào **face\_recognition** để dễ sử dụng hơn) nhằm tạo ra các **face embeddings.**

Trong phần này file **encode\_faces.py** được chạy để lưu các encodings và names (nếu cần ID thì bổ sung). Các bạn xem thêm file đó để hiểu hơn, mình có chú thích rất rõ từng phần. Các encodings và names được lưu ra file **encodings.pickle**.

**Bước 3. Nhận dạng khuôn mặt trong ảnh và trích xuất thông tin từ trên firebase về**



Hình 3.1.3 Ảnh đầu ra sau khi nhận diện

Sau khi chúng ta đã có các encodings từ datasets (lấy được qua pre-trained model, thông qua dlib và face\_recognition) chúng ta có thể bắt đầu thực hiện face recognition được rồi.

Chạy file **recognize\_faces\_image.py** để nhận diện khuôn mặt trong ảnh.

Chú ý:

Nếu muốn chạy face recognition trên CPU hay các thiết bị nhúng như Raspberry thì chọn detection method là **hog** ở file **recognize\_faces\_image.py**, còn ban đầu khi lấy encodings từ dataset chúng ta vẫn có thể để **cnn** (chạy lâu hơn nhưng chính xác hơn để phát hiện khuôn mặt).

# **3.3. Dữ liệu**

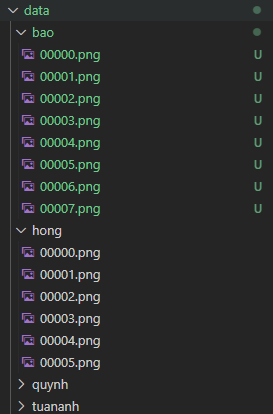
Dữ liệu đóng vai trò rất quan trọng trong các bài sử dụng học sâu nói chung hay chính xác là mạng nơron. Chất lượng, độ tin cậy, tính sẵn có và phù hợp của dữ liệu được sử dụng để xây dựng mô hình giúp nâng cao độ chính xác cho đầu ra của các bài toán. Kể cả với những mô hình đơn giản cũng có thể đạt được những kết quả tốt nếu như dữ liệu đầu vào đã được xử lý tốt, nắm giữ các thông tin quan trọng. Cùng với đó, các mô hình tuy rằng tốt có thể sẽ không cho ta các kết quả mong muốn nếu dữ liệu đầu vào phức tạp, rắc rối và chứa nhiều dữ liệu nhiễu.

Việc xử lý dữ liệu bắt đầu bằng việc thu thập và phân tích dữ liệu, sau đó là bước tiền xử lý. Dữ liệu sau khi qua bước tiền xử lý được đưa vào mô hình. Cuối cùng, dữ liệu đầu ra của mạng nơron qua bước hậu xử lý, ở bước này sẽ thực hiện biến đổi kết quả trả về của mạng nơron sang dạng hiểu được yêu cầu của bài toán. Sau đây, ta xem quá trình xử lý dữ liệu.

## **3.3.1 Mô tả dữ liệu đầu vào**

Tập dữ liệu được sử dụng ở đây là ảnh các đối tượng muốn nhận dạng.

Bộ dữ liệu bao gồm hai phần chính: **tên tệp ảnh của đối tượng** và **ảnh của đối tượng**. Mỗi đối tượng được chụp từ 5 đến 7 ảnh, với tên tệp được đặt theo định dạng xxxxx.png, trong đó x tương ứng với các số từ 1 đến 9. Cách đặt tên này giúp phân biệt từng ảnh của đối tượng một cách rõ ràng và thuận tiện cho việc quản lý dữ liệu.



Hình 3.3.1 Hình ảnh dữ liệu của các đối tượng cần nhận dạng

## **3.2.2 Xử lý dữ liệu**

Tiền xử lý dữ liệu là một bước rất quan trọng trong việc giải quyết các vấn đề về học máy. Hầu hết các bộ dữ liệu được sử dụng trong các vấn đề liên quan đến học máy cần được xử lý, làm sạch và chuyển đổi trước khi đào tạo các thuật toán học máy trên các bộ dữ liệu đó. Các kỹ thuật tiền xử lý dữ liệu hiện tại bao gồm: Xử lý dữ liệu bị thiếu, mã hóa các biến phân loại, chuẩn hóa dữ liệu, chia tỷ lệ dữ liệu, v.v.

**Transfer learning cho dữ liệu ảnh đầu vào**

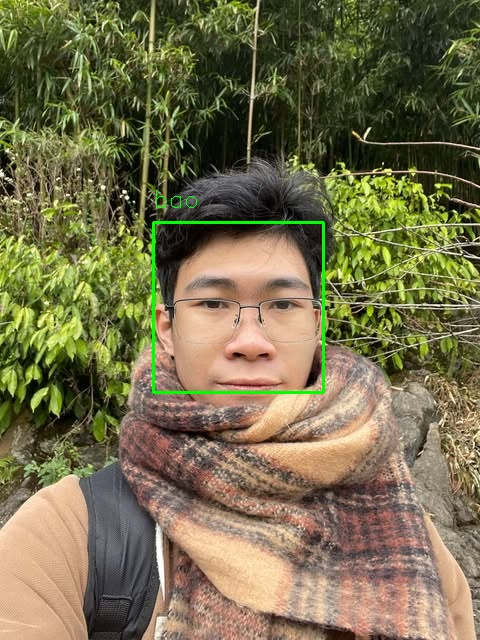
Hiện nay, các mô hình lớn như VGG, ResNet, và Inception đã được huấn luyện trên các tập dữ liệu rất lớn và đạt được kết quả xuất sắc trong nhiều bài toán xử lý ảnh. Thay vì huấn luyện lại từ đầu, chúng ta có thể tận dụng những mô hình này để giải quyết bài toán khi dữ liệu còn hạn chế, điều này giúp giảm thiểu thời gian và tài nguyên tính toán.

Kỹ thuật này được gọi là Transfer Learning. Đây là một phương pháp giúp tận dụng kiến thức đã học được từ các bài toán trước để giải quyết các bài toán mới, đặc biệt là khi tập dữ liệu có nhãn còn thiếu hoặc hạn chế. Trong trường hợp này, chúng ta sẽ sử dụng mô hình đã được huấn luyện để thu thập các đặc trưng của ảnh trong tập dữ liệu của mình, thay vì huấn luyện từ đầu.

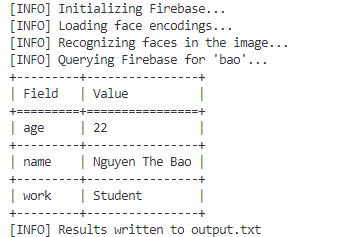
Đối với bài toán nhận diện khuôn mặt, chúng em sẽ sử dụng Mô hình nhận diện khuôn mặt (Face Recognition Model). Mô hình này sẽ được áp dụng để trích xuất đặc trưng khuôn mặt từ ảnh trong tập dữ liệu mẫu, giúp chúng ta nhận diện và phân loại khuôn mặt một cách chính xác hơn dù dữ liệu có phần hạn chế.

Việc áp dụng Transfer Learning trong mô hình nhận diện khuôn mặt giúp tận dụng những đặc trưng mạnh mẽ đã được học từ các mô hình lớn, đồng thời giảm thiểu vấn đề thiếu dữ liệu huấn luyện.

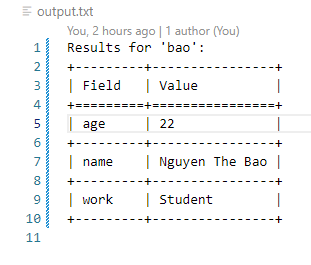
**3.4. Kết quả thực nghiệp**

****

Hình 3.3.1 kết quả nhận diện đối tượng

****

Hình 3.3.2 Xuất thông tin ra Teminal

****

Hình 3.3.3 Xuất thông tin ra file output.txt

**Kết luận**

Hệ thống nhận dạng đối tượng và trích xuất thông tin đối tượng trong ảnh sử dụng mô hình nhận diện khuôn mặt (Face Recognition Model) mang lại hiệu quả cao trong việc phân tích và xử lý hình ảnh. Quá trình này bắt đầu từ việc thu thập dữ liệu hình ảnh từ các nguồn như camera, sau đó trích xuất đặc trưng của đối tượng trong ảnh và lưu trữ chúng dưới dạng file dữ liệu. Mô hình nhận diện khuôn mặt sẽ xử lý các đặc trưng này để nhận dạng và phân loại đối tượng, từ đó giúp trích xuất thông tin liên quan đến đối tượng nhận diện. Hệ thống này không chỉ đảm bảo độ chính xác cao trong việc nhận diện mà còn giúp tối ưu hóa quá trình quản lý và sử dụng thông tin đối tượng, hỗ trợ các ứng dụng trong nhiều lĩnh vực như bảo mật, giám sát và xác thực người dùng.

**Hướng phát triển:**

1. Cải tiến mô hình: Nâng cao độ chính xác và khả năng nhận diện các khuôn mặt trong điều kiện ánh sáng kém, góc chụp không thuận lợi hoặc các thay đổi về hình dáng khuôn mặt như độ tuổi hoặc thay đổi kiểu tóc.
2. Tích hợp trí tuệ nhân tạo (AI): Phát triển các hệ thống nhận diện đa đối tượng hoặc các mô hình có khả năng nhận diện các đối tượng khác ngoài khuôn mặt, như nhận diện các vật thể hoặc hành vi trong ảnh.
3. Ứng dụng trong các lĩnh vực mới: Mở rộng ứng dụng của hệ thống trong các lĩnh vực như chăm sóc sức khỏe (nhận diện bệnh lý qua hình ảnh y tế), an ninh công cộng (nhận diện trong đám đông) hoặc bán lẻ (quản lý khách hàng và theo dõi hành vi người tiêu dùng).

**Ưu điểm:**

**Độ chính xác cao**: Mô hình nhận diện khuôn mặt có khả năng nhận diện đối tượng chính xác trong nhiều điều kiện khác nhau.

**Tiết kiệm thời gian**: Quá trình nhận diện tự động giúp giảm thời gian cần thiết để nhận dạng đối tượng thủ công.

**Tính linh hoạt**: Hệ thống có thể áp dụng trong nhiều lĩnh vực khác nhau, từ bảo mật, giám sát đến ứng dụng trong kinh doanh.

**Tiết kiệm chi phí**: So với các phương pháp truyền thống như kiểm tra thủ công hoặc sử dụng thẻ, hệ thống nhận diện khuôn mặt giúp giảm chi phí dài hạn và tăng tính hiệu quả.

**Nhược điểm:**

**Yêu cầu tài nguyên phần cứng cao**: Các mô hình nhận diện khuôn mặt đòi hỏi phần cứng mạnh mẽ, đặc biệt là khi xử lý lượng dữ liệu lớn hoặc trong thời gian thực.

**Độ chính xác phụ thuộc vào chất lượng ảnh**: Nếu chất lượng hình ảnh không cao (ví dụ như ánh sáng kém, góc chụp không tốt), mô hình có thể gặp khó khăn trong việc nhận diện chính xác.

**Vấn đề bảo mật và quyền riêng tư**: Việc thu thập và lưu trữ dữ liệu khuôn mặt có thể gặp phải vấn đề về bảo mật và quyền riêng tư, yêu cầu tuân thủ các quy định về bảo vệ dữ liệu cá nhân.

**Khả năng bị giả mạo**: Mặc dù mô hình nhận diện khuôn mặt rất hiệu quả, nhưng trong một số trường hợp, công nghệ này có thể bị tấn công bằng các phương pháp giả mạo như sử dụng ảnh chụp hoặc video giả.

Tóm lại, hệ thống nhận diện khuôn mặt là một công cụ mạnh mẽ với nhiều tiềm năng phát triển, nhưng cũng cần phải chú ý đến các yếu tố như bảo mật và các hạn chế kỹ thuật trong quá trình ứng dụng thực tế.

# **Tài liệu tham khảo**

1) Tran Van Huy 2017, Face Recognition là gì?, 30/11/2024 form [Face Recognition là gì? – Tran Van Huy – Artificial Intellegence](https://huytranvan2010.github.io/Face-Recognition/#:~:text=Face%20recognition%20%28nh%E1%BA%ADn%20di%E1%BB%87n%20khu%C3%B4n%20m%E1%BA%B7t%29%20l%C3%A0%20qu%C3%A1,ch%C3%ADnh%20x%C3%A1c%20c%E1%BB%A7a%20ph%C6%B0%C6%A1ng%20ph%C3%A1p%20n%C3%A0y%20kh%C3%B4ng%20cao.)

2) Adrian Rosebrock 2018 Face recognition with OpenCV, Python, and deep learning 24/11/2024 form [Face recognition with OpenCV, Python, and deep learning - PyImageSearch](https://pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/)

3) Machine Learning cơ bản (machinelearningcoban.com)

4) Deep Learning cơ bản | Nguyễn Thanh Tuấn (nttuan8.com)

5) Machine Learning | Coursera (coursera.com)

6) Deep Learning Specialization | Coursera (coursera.com)

7) Andrew Ng, Convolutional Neural Networks, Convolutional Neural Networks (coursera.com)