

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN 1**

**BỘ MÔN HỆ CƠ SỞ DỮ LIỆU ĐA PHƯƠNG TIỆN**

---



**XÂY DỰNG HỆ CSDL LƯU TRỮ  
VÀ TÌM KIẾM MẶT NGƯỜI**

**Giảng viên hướng dẫn : Nguyễn Đình Hóa**

**Họ và tên sinh viên : Đặng Ngọc Anh B21DCCN141**

**Hoàng Đức Huyền B21DCCN063**

**Phạm Ngọc Trung B21DCCN734**

**Lớp : 04**

**Nhóm : 29**

**Hà Nội – 2025**

## LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành và sâu sắc đến Thầy Nguyễn Đình Hóa – giảng viên môn Hệ cơ sở dữ liệu đa phương tiện đã tận tình giảng dạy, hướng dẫn và truyền đạt cho em những kiến thức quý báu trong suốt quá trình học tập và thực hiện đề tài này.

Nhờ sự hướng dẫn tận tâm và chỉ bảo nhiệt tình của thầy, em đã có cơ hội tiếp cận và ứng dụng những kiến thức lý thuyết vào việc xây dựng một hệ thống cơ sở dữ liệu lưu trữ và tìm kiếm ảnh khuôn mặt – một chủ đề có tính ứng dụng thực tiễn cao trong lĩnh vực công nghệ thông tin ngày nay.

Em cũng xin cảm ơn các thầy cô trong khoa Công nghệ thông tin cùng các bạn học đã hỗ trợ và đóng góp ý kiến để em hoàn thiện báo cáo này tốt hơn.

Tuy đã cố gắng hoàn thành đề tài với tinh thần nghiêm túc, nhưng do thời gian và kiến thức còn hạn chế, bài báo cáo chắc chắn không tránh khỏi những thiếu sót. Em kính mong Thầy góp ý để em có thể rút kinh nghiệm và hoàn thiện hơn trong các dự án sau này.

# LỜI MỞ ĐẦU

Trong bối cảnh cuộc cách mạng công nghiệp 4.0 đang diễn ra mạnh mẽ, việc xử lý và khai thác hiệu quả các loại dữ liệu đa phương tiện, đặc biệt là dữ liệu hình ảnh, ngày càng trở nên quan trọng và cần thiết. Một trong những ứng dụng nổi bật của lĩnh vực này là hệ thống nhận diện và tìm kiếm khuôn mặt – vốn đã và đang được triển khai rộng rãi trong an ninh, y tế, giáo dục và các nền tảng công nghệ hiện đại.

Với mục tiêu vận dụng kiến thức đã học để giải quyết một bài toán thực tế, đề tài “Xây dựng hệ cơ sở dữ liệu lưu trữ và tìm kiếm mặt người” được lựa chọn làm nội dung cho báo cáo môn học Hệ cơ sở dữ liệu đa phương tiện. Đề tài tập trung vào việc thu thập dữ liệu ảnh khuôn mặt, trích xuất các thuộc tính đặc trưng, xây dựng mô hình lưu trữ và triển khai chức năng tìm kiếm ảnh dựa trên độ tương đồng giữa các khuôn mặt.

Báo cáo này sẽ trình bày toàn bộ quá trình xây dựng hệ thống, từ bước thu thập và xử lý dữ liệu đến trích xuất đặc trưng và xây dựng thuật toán tìm kiếm, đồng thời đánh giá hiệu quả và tính thực tiễn của hệ thống đã triển khai.

## Phân chia công việc

Thành viên	Công việc
Đặng Ngọc Anh	Tìm nguồn ảnh. Xây dựng một bộ thuộc tính, hệ thống, hệ CSDL lưu trữ. Viết báo cáo
Phạm Ngọc Trung	Tìm nguồn ảnh. Xây dựng bộ thuộc tính và hệ thống
Hoàng Đức Huyền	

## Mục lục

Chương 1: Mở đầu .....	7
1.1 Lý do chọn đề tài .....	7
1.2 Mục tiêu nghiên cứu .....	7
1.3 Phạm vi và giới hạn đề tài .....	8
Chương 2: Cơ sở lý thuyết và công nghệ sử dụng .....	8
2.1 Cơ sở lý thuyết .....	8
2.1.1 Cơ sở dữ liệu đa phương tiện (Multimedia Database) .....	8
2.1.2 Nhận diện khuôn mặt (Face Recognition).....	8
2.1.3 Bộ thuộc tính đặc trưng và giá trị thông tin trong nhận diện khuôn mặt .....	11
2.1.4 Phép đo khoảng cách Euclidean .....	18
2.1.5 Tìm kiếm tương tự với FAISS.....	19
2.2 Công nghệ và thư viện sử dụng .....	19
2.3 Mô hình lưu trữ ảnh và đặc trưng khuôn mặt.....	19
2.4 Quy trình hoạt động của hệ thống.....	20
Chương 3: Phân tích và thiết kế hệ thống .....	20
3.1 Mô hình tổng quát của hệ thống .....	20
3.1.1 Sơ đồ khối của hệ thống .....	21
3.2 Phân tích yêu cầu hệ thống.....	21
3.2.1 Yêu cầu chức năng .....	21
3.2.2 Yêu cầu phi chức năng .....	21
3.3 Thiết kế cơ sở dữ liệu.....	22
3.4 Thiết kế thuật toán tìm kiếm ảnh tương tự.....	22
3.4.1 Quy trình tìm kiếm.....	22
3.4.2 Cách tính độ tương đồng.....	22
3.4.3 Ưu điểm của FAISS .....	23
3.5 Ưu điểm của hệ thống .....	23
CHƯƠNG 4. DEMO VÀ ĐÁNH GIÁ HỢI HIỆU .....	23
4.1 Demo hệ thống .....	23

4.1.1 Giao diện hệ thống.....	24
4.1.2 Giao diện hiển thị kết quả.....	25
4.2 Đánh giá kết quả .....	25
4.3 Hướng phát triển .....	26
CHƯƠNG 5: KẾT LUẬN .....	26
5.1. Tóm tắt nội dung đã thực hiện.....	26
5.2. Kết quả đạt được .....	27
5.3. Hạn chế .....	27
5.4. Hướng phát triển .....	27

# Chương 1: Mở đầu

## 1.1 Lý do chọn đề tài

Trong thời đại số hóa và phát triển mạnh mẽ của trí tuệ nhân tạo (AI), công nghệ nhận diện khuôn mặt đã trở thành một trong những ứng dụng quan trọng và phổ biến nhất. Việc nhận diện và tìm kiếm ảnh khuôn mặt đóng vai trò then chốt trong nhiều lĩnh vực như an ninh – giám sát, quản lý nhân sự, dịch vụ khách hàng, thương mại điện tử, và các ứng dụng xã hội khác. Đặc biệt, với sự bùng nổ của các thiết bị di động và camera số, lượng ảnh khuôn mặt được thu thập ngày càng nhiều, đòi hỏi các hệ thống lưu trữ và tìm kiếm phải đáp ứng được yêu cầu về tốc độ, độ chính xác và khả năng mở rộng.

Đề tài “Xây dựng hệ cơ sở dữ liệu lưu trữ và tìm kiếm mặt người” được chọn nhằm giải quyết bài toán lưu trữ và tìm kiếm ảnh mặt người nữ thuộc nhiều độ tuổi trong một tập dữ liệu đa dạng, đồng thời xây dựng một hệ thống tìm kiếm ảnh mặt người nhanh và chính xác dựa trên các thuộc tính đặc trưng khuôn mặt được trích xuất từ ảnh. Đây là một chủ đề thiết thực, có tính ứng dụng cao, đồng thời giúp người học làm quen với các công nghệ xử lý ảnh, học máy, và quản trị cơ sở dữ liệu đa phương tiện.

## 1.2 Mục tiêu nghiên cứu

Mục tiêu chính của đề tài gồm:

- Thu thập và xây dựng bộ dữ liệu ảnh mặt người nữ với số lượng ít nhất 200 ảnh, thuộc nhiều độ tuổi khác nhau, chuẩn hóa kích thước và tỉ lệ ảnh để đảm bảo đồng nhất dữ liệu đầu vào.
- Xây dựng bộ thuộc tính nhận diện khuôn mặt hiệu quả dựa trên vector embedding trích xuất từ mô hình học sâu, nhằm biểu diễn đặc trưng khuôn mặt của từng cá nhân.
- Thiết kế và triển khai hệ thống tìm kiếm ảnh mặt người, với đầu vào là ảnh mới và đầu ra là 3 ảnh giống nhất trong cơ sở dữ liệu, sắp xếp theo thứ tự giảm dần về độ tương đồng.
- Đánh giá hiệu quả của hệ thống dựa trên kết quả tìm kiếm thực nghiệm.

## 1.3 Phạm vi và giới hạn đề tài

Đề tài tập trung vào:

- Xây dựng hệ cơ sở dữ liệu và hệ thống tìm kiếm ảnh mặt người thuộc giới tính nữ, nhằm đảm bảo tính đa dạng về độ tuổi và đặc điểm khuôn mặt.
- Ảnh sử dụng là ảnh tĩnh, kích thước và tỉ lệ khung hình được chuẩn hóa (ví dụ 160x160 pixel).
- Sử dụng mô hình học sâu có sẵn để trích xuất vector embedding, tập trung vào độ chính xác và tốc độ tìm kiếm.
- Cơ sở dữ liệu sử dụng MySQL để lưu trữ ảnh và vector embedding, kết hợp với thư viện FAISS để tìm kiếm nhanh các vector đặc trưng.
- Hệ thống chưa xử lý ảnh đa người, video hay các trường hợp ánh sáng phức tạp.
- Giao diện hệ thống ở mức đơn giản, chủ yếu phục vụ mục đích nghiên cứu và demo chức năng tìm kiếm.

## Chương 2: Cơ sở lý thuyết và công nghệ sử dụng

### 2.1 Cơ sở lý thuyết

#### 2.1.1 Cơ sở dữ liệu đa phương tiện (Multimedia Database)

Cơ sở dữ liệu đa phương tiện là loại CSDL có khả năng lưu trữ và quản lý dữ liệu phi cấu trúc như ảnh, âm thanh, video, văn bản... Trong đề tài này, ảnh khuôn mặt được lưu trữ cùng với các thuộc tính đặc trưng dạng vector số học để hỗ trợ tìm kiếm theo nội dung (content-based retrieval).

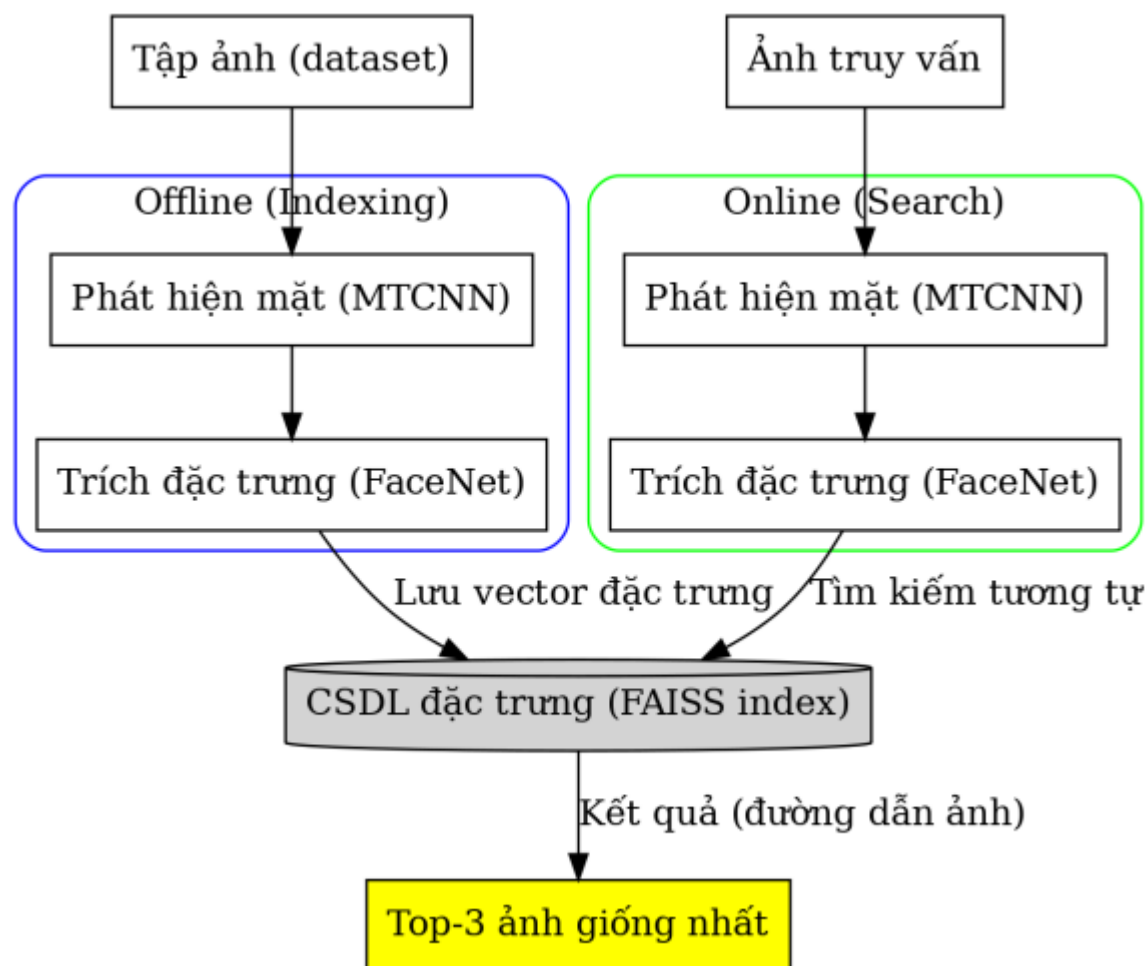
#### 2.1.2 Nhận diện khuôn mặt (Face Recognition)

**Giới thiệu:** Hệ thống tìm kiếm ảnh khuôn mặt cho phép so sánh một ảnh khuôn mặt đầu vào với một cơ sở dữ liệu ảnh để tìm ra những khuôn mặt giống nhất. Công nghệ này kết hợp phát hiện khuôn mặt (nhận diện và cắt vùng mặt trong ảnh), trích xuất đặc trưng bằng mô hình học sâu để biểu diễn khuôn mặt dưới dạng vector, và tìm kiếm xấp xỉ gần nhất trong không gian vector để truy vấn ảnh



tương tự một cách nhanh chóng. Dưới đây là sơ đồ khối minh họa các bước chính của hệ thống cùng với mô tả chi tiết từng bước xử lý.

**Sơ đồ khối hệ thống tìm kiếm ảnh khuôn mặt (đầu vào → xử lý → đầu ra)**



Hình 1: Sơ đồ khối của hệ thống tìm kiếm ảnh khuôn mặt. Hệ thống bao gồm các bước xử lý tuần tự từ ảnh đầu vào đến kết quả đầu ra. Ảnh đầu vào trước tiên được đưa qua bước phát hiện khuôn mặt (Face Detection) bằng MTCNN để xác định và cắt lấy vùng khuôn mặt trong ảnh. Tiếp theo, vùng mặt được đưa vào mô hình FaceNet (mạng InceptionResnetV1 đã huấn luyện sẵn trên tập dữ liệu VGGFace2) để trích xuất một vector đặc trưng 512 chiều đại diện cho khuôn mặt đó. Tất cả các vector đặc trưng của ảnh trong cơ sở dữ liệu sẽ được lưu trữ (có thể dưới dạng tệp .npy hoặc lưu trong cơ sở dữ liệu quan hệ kèm thông tin đường dẫn ảnh) và đồng thời được lập chỉ mục bằng công cụ FAISS (Facebook AI Similarity Search) để phục vụ tìm kiếm nhanh. Khi có một ảnh truy vấn mới, hệ thống tính toán vector đặc trưng cho khuôn mặt trong ảnh đó tương tự các bước trên, rồi sử dụng FAISS để tìm kiếm các vector gần nhất trong CSDL đặc trưng

*đã lập chỉ mục . Kết quả trả về là top-3 ảnh có khuôn mặt giống nhất (kèm theo đường dẫn hoặc thông tin tương ứng) được trích xuất từ cơ sở dữ liệu.*

## **Quy trình trích xuất, lưu trữ và tìm kiếm đặc trưng chi tiết**

Quy trình xử lý dữ liệu và tìm kiếm trong hệ thống được thực hiện qua các bước chính sau:

**1.Phát hiện khuôn mặt:** Sử dụng bộ phát hiện MTCNN (Multi-task Cascaded Convolutional Networks) để xác định khuôn mặt trong ảnh đầu vào. MTCNN trả về tọa độ bounding box của khuôn mặt, và hệ thống sẽ cắt vùng ảnh theo tọa độ này để thu được khuôn mặt đã được căn chỉnh . Bước phát hiện giúp cô lập khuôn mặt và loại bỏ nền hoặc phần thừa, chuẩn bị cho bước nhận dạng tiếp theo.

**2.Trích xuất đặc trưng (embedding):** Vùng mặt cắt được đưa vào mô hình FaceNet để trích xuất vector đặc trưng có kích thước cố định (thường là 512 chiều) biểu diễn cho khuôn mặt đó . Vector này chính là embedding – một biểu diễn số học của các đặc điểm khuôn mặt, sao cho các khuôn mặt giống nhau sẽ có embedding ở gần nhau trong không gian vector . Mô hình FaceNet đã được huấn luyện để tạo ra embedding sao cho khoảng cách giữa các vector tương ứng với mức độ khác biệt khuôn mặt: cùng một người sẽ cho vector rất gần, trong khi người khác biệt sẽ cho vector xa hơn.

**3.Lưu trữ vector đặc trưng:** Tất cả các vector đặc trưng trích xuất từ tập ảnh dữ liệu (ảnh đầu vào đã biết) được lưu trữ vào một cơ sở dữ liệu để phục vụ tìm kiếm. Trong mã nguồn cung cấp, các vector này được lưu trong một mảng NumPy (tệp .npy ) kèm theo danh sách các đường dẫn tới ảnh gốc tương ứng . (Ngoài ra, tùy ứng dụng, có thể lưu các vector vào một bảng trong cơ sở dữ liệu quan hệ như MySQL cùng với metadata về ảnh để dễ tra cứu .) Sau khi lưu trữ, hệ thống sẽ xây dựng chỉ mục FAISS trên tập hợp các vector này. FAISS là thư viện của Facebook AI chuyên tối ưu cho việc tìm kiếm vector tương tự một cách hiệu quả – việc lập chỉ mục bằng FAISS cho phép tìm láng giềng gần nhất trong tập vector rất nhanh chóng, ngay cả khi cơ sở dữ liệu có hàng triệu vector .

**4.Tìm kiếm ảnh tương tự:** Khi có một ảnh truy vấn mới, hệ thống thực hiện lại bước phát hiện khuôn mặt và trích xuất vector đặc trưng cho khuôn mặt

trong ảnh truy vấn đó. Tiếp theo, vector đặc trưng của ảnh truy vấn được so sánh với các vector trong CSDL đặc trưng đã lập chỉ mục bằng FAISS. Cụ thể, FAISS sẽ tìm ra những vector lân cận gần nhất (nearest neighbors) trong không gian theo tiêu chí khoảng cách (ví dụ khoảng cách Euclidean) so với vector truy vấn. Kết quả trả về là danh sách các ảnh có khuôn mặt giống nhất với ảnh truy vấn – thông thường hệ thống sẽ trả ra top 3 ảnh tương tự nhất cùng với khoảng cách hoặc độ tương đồng tương ứng. Dựa vào kết quả này, ta có thể hiển thị các ảnh tương tự (ví dụ hiển thị hình ảnh hoặc đường dẫn tới ảnh trong cơ sở dữ liệu) cho người dùng.

### 2.1.3 Bộ thuộc tính đặc trưng và giá trị thông tin trong nhận diện khuôn mặt

#### **Phát hiện và Chuẩn hóa Khuôn Mặt (Face Detection & Normalization)**

Hệ thống sử dụng module `face_detector.py` với mô hình MTCNN để phát hiện khuôn mặt trong ảnh đầu vào. Cụ thể, MTCNN (Multi-Task Cascaded Convolutional Networks) sẽ xác định vị trí khuôn mặt và các điểm đặc trưng trên ảnh. Code đọc ảnh bằng PIL, chuyển sang RGB rồi nếu ảnh quá lớn thì thu nhỏ lại (giới hạn kích thước lớn nhất 800px) để tăng tốc xử lý. Sau đó, `detector.detect_faces(pixels)` được gọi để lấy bounding box khuôn mặt. Kết quả là tọa độ (x1, y1) và (x2, y2) bao quanh khuôn mặt, dùng để crop khuôn mặt từ ảnh gốc. Khuôn mặt được cắt sau đó được resize về kích thước chuẩn (mặc định 160×160 pixel) trước khi đưa vào bước trích xuất đặc trưng. Việc chuẩn hóa này đảm bảo khuôn mặt đầu vào cho mô hình nhận dạng có kích thước nhất quán (160×160) và tập trung vào vùng mặt chính, loại bỏ nền không cần thiết. Điều này phù hợp với yêu cầu của mô hình Facenet, vốn được huấn luyện trên ảnh khuôn mặt 160×160 và khuyến nghị ảnh vào đã được cắt đúng vùng mặt bằng MTCNN.

#### **Trích Xuất Đặc Trưng Khuôn Mặt Bằng Mô Hình Sâu (Deep Feature Extraction)**

Module `embedder.py` đảm nhiệm việc chuyển ảnh khuôn mặt đã chuẩn hóa thành vector đặc trưng (face embedding) bằng mô hình học sâu InceptionResnetV1 từ thư viện `facenet_pytorch`. Khi một ảnh khuôn mặt 160×160×3 (RGB) được đưa vào, code sẽ chuẩn hóa giá trị điểm ảnh từ [0,255] về [0,1] rồi về khoảng [-1,1] (bằng cách trừ 0.5 rồi chia 0.5) – đúng theo tiền xử lý chuẩn của mạng Facenet.

Ảnh sau đó được chuyển thành tensor PyTorch định dạng (C,H,W) và thêm chiều batch . Mô hình InceptionResnetV1 (đã tải trọng số pre-trained trên VGGFace2) được load sẵn ở chế độ eval , và được dùng để suy luận ra vector đặc trưng cho khuôn mặt. Cụ thể, lệnh `model(face_tensor)` trả về một tensor 2 chiều kích thước (1, d) – với d là số chiều của vector đặc trưng, sau đó được chuyển thành NumPy array . Theo tài liệu Facenet, mô hình InceptionResnetV1 khi dùng cho nhận dạng sẽ xuất ra vector 512 chiều đại diện cho khuôn mặt (mặc định mạng tạo embedding 512 chiều) . Mỗi embedding là một mảng gồm 512 số thực (float32) mô tả các đặc trưng sâu của khuôn mặt đó.

**Định dạng và kích thước:** Vector đặc trưng nhận được là một mảng NumPy 1 chiều độ dài 512. Trong code, vector này được trả về từ hàm `get_embedding` dưới dạng `numpy.ndarray float32` . Mỗi phần tử trong vector không tương ứng với một đặc trưng hữu hình cụ thể nào của khuôn mặt (như mắt, mũi) mà là kết quả của nhiều tầng tích chập và kết nối phức tạp trong mạng neural. Nói cách khác, embedding là biểu diễn trích xuất đặc trưng cấp cao: nó mã hóa thông tin nhận dạng của khuôn mặt trong không gian 512 chiều. Các giá trị tuyệt đối của từng phần tử không có ý nghĩa trực quan riêng lẻ, nhưng toàn bộ vector tạo thành một “dấu vân tay” duy nhất của khuôn mặt đó trong không gian đặc trưng.

## **Ý Nghĩa của Vector Đặc Trưng (Face Embedding Vector)**

Vector embedding 512 chiều này chính là bộ thuộc tính đại diện cho khuôn mặt của một người. Mục tiêu của Facenet là ánh xạ ảnh khuôn mặt vào một không gian vector sao cho: những ảnh cùng một người sẽ có vector nằm gần nhau, còn ảnh của những người khác nhau sẽ nằm xa nhau trong không gian đó . Nhờ được huấn luyện trên tập dữ liệu khuôn mặt khổng lồ (như VGGFace2, chứa hàng nghìn người), mô hình học sâu đã học cách nhận diện các đặc trưng quan trọng nhất để phân biệt danh tính. Các đặc trưng này bao gồm hình dạng cấu trúc gương mặt, khoảng cách tương đối giữa các bộ phận (mắt, mũi, miệng), cũng như các nét riêng khác – và quan trọng là chúng ít bị ảnh hưởng bởi các yếu tố nhiễu như pose (góc độ khuôn mặt), ánh sáng, biểu cảm.

**Tính ổn định trước biến đổi:** So với việc dùng pixel thô hoặc các đặc trưng thủ công, embedding sâu cho tính ổn định cao trước các thay đổi về tư thế, ánh sáng hay biểu cảm khuôn mặt. Ví dụ, dù một người cười hay nghiêm mặt, quay nghiêng hay chính diện, ở ngoài trời nắng hay trong phòng tối, thì vector

embedding của họ vẫn sẽ giữ tương đồng cao, vì mạng đã học được cách trích xuất những yếu tố bất biến quan trọng cho nhận dạng. Điều này đạt được nhờ hàm loss đặc biệt (triplet loss) trong quá trình huấn luyện Facenet, đảm bảo khoảng cách giữa các ảnh của cùng một người (anchor và positive) được kéo gần, còn ảnh của người khác (negative) được đẩy ra xa. Kết quả, vector đặc trưng tập trung thể hiện danh tính cốt lõi của gương mặt, loại bỏ hầu hết ảnh hưởng của môi trường chụp.

Tính phân biệt cao: Mỗi người sẽ có một vector đặc trưng khác biệt trong không gian 512 chiều. Nhờ độ phân giải cao của không gian này, hệ thống có khả năng phân biệt hàng ngàn, thậm chí hàng triệu khuôn mặt khác nhau mà vẫn duy trì được khoảng cách đủ lớn giữa các lớp (mỗi lớp tương ứng một người). Những vector của hai người khác nhau hiếm khi trùng gần nhau, vì nếu mạng được huấn luyện tốt, nó sẽ học cách nhấn mạnh vào các chi tiết đặc trưng riêng của từng người (ví dụ: cấu trúc xương mặt, hình dáng mắt, v.v). Do đó, embedding đóng vai trò như một mã nhận dạng duy nhất cho mỗi khuôn mặt – tương tự như mã số định danh, nhưng được rút trích từ chính dữ liệu ảnh một cách tự động.

## **Sử Dụng Embedding cho So Khớp và Nhận Dạng Khuôn Mặt**

Vector embedding là nền tảng cho việc so khớp và nhận dạng ảnh khuôn mặt trong hệ thống này. Thay vì so sánh trực tiếp ảnh với ảnh (vốn phức tạp và chịu ảnh hưởng của nhiều yếu tố thô như độ phân giải, ánh sáng), hệ thống so sánh vector đặc trưng. Cụ thể, khi cần tìm kiếm hoặc nhận diện, ảnh đầu vào sẽ được qua các bước: phát hiện mặt → trích xuất embedding như trên, gọi là query embedding. Sau đó, so sánh query embedding này với tập các embedding đã lưu của ảnh trong cơ sở dữ liệu. Việc so sánh thường dựa trên khoảng cách giữa các vector trong không gian 512 chiều – ví dụ khoảng cách Euclid (L2) hoặc cosine. Hai ảnh được xem là cùng một người nếu khoảng cách giữa hai vector embedding của chúng nhỏ hơn một ngưỡng nhất định.

Trong mã nguồn, chức năng so khớp được hiện thực rõ ở các file `search.py` và `search_only.py`. Hàm `search(image_path, index, image_paths, top_k=3)` sẽ: (1) sử dụng `extract_face` để lấy khuôn mặt từ ảnh truy vấn; (2) dùng `get_embedding` để lấy vector đặc trưng cho khuôn mặt đó; (3) dùng FAISS index đã xây dựng từ trước để tìm các vector gần nhất (tương tự nhất) trong cơ sở dữ liệu so với query

vector . Kết quả trả về là danh sách các ảnh có embedding gần nhất cùng khoảng cách tương ứng . Trong `search_only.py` (ứng dụng GUI), quy trình tương tự: khi người dùng chọn một ảnh, hệ thống cắt mặt, tính embedding rồi gọi `index.search(query_embedding, 5)` để lấy ra danh sách 5 gương mặt giống nhất . Kết quả top 3 (bỏ qua chính ảnh truy vấn) sẽ được hiển thị lên giao diện . 13 14 15 15 16 17 18 19 20 2 Như vậy, giá trị thông tin của vector embedding nằm ở chỗ: nó cho phép định lượng mức độ giống nhau giữa hai khuôn mặt bằng một con số (khoảng cách vector). Vector của cùng một người sẽ rất gần nhau, còn của hai người khác sẽ xa hơn. Điều này làm cơ sở để thực hiện truy vấn “tìm ảnh khuôn mặt tương tự” hoặc “xác định danh tính ảnh này” một cách nhanh chóng và hiệu quả. Toàn bộ quá trình nhận dạng được đơn giản hóa thành bài toán tìm láng giềng gần nhất trong không gian vector – một bài toán mà máy tính xử lý rất nhanh, thay vì phải so khớp từng điểm ảnh phức tạp.

## **Lưu Trữ Embedding và Tăng Tốc Tìm Kiếm với FAISS**

Lưu trữ vector đặc trưng: Sau khi trích xuất, các vector embedding sẽ được lưu lại để phục vụ tìm kiếm sau này. Có nhiều cách lưu trữ tùy theo ứng dụng: - Lưu dưới dạng file nén ( `.npy` ): Code ví dụ dùng NumPy lưu mảng các embedding thành file nhị phân `.npy` , và song song lưu danh sách đường dẫn ảnh tương ứng vào file văn bản . Cách này tiện lợi khi số lượng ảnh vừa phải và có thể tải tất cả embedding vào bộ nhớ khi tìm kiếm. - Lưu trong CSDL (MySQL BLOB): Hệ thống cũng hỗ trợ lưu embedding vào database. Mỗi vector (`float32[512]`) được chuyển thành chuỗi byte và lưu vào trường kiểu BLOB trong bảng SQL cùng với đường dẫn ảnh . Khi cần tải, chương trình đọc BLOB từ DB rồi dùng `np.frombuffer` để phục hồi thành mảng `float32` . Lưu trong CSDL giúp dễ dàng quản lý, mở rộng, và tích hợp với các thông tin khác (vd: ID người, thông tin kèm theo).

Tăng tốc tìm kiếm với FAISS: Để tìm ảnh tương tự nhanh chóng trong kho lớn các embedding, hệ thống sử dụng FAISS (Facebook AI Similarity Search) – một thư viện tối ưu cho tìm kiếm vector gần nhất. FAISS cho phép xây dựng chỉ mục (index) hiệu quả trên tập các vector, hỗ trợ tìm top-k lân cận rất nhanh nhờ các cấu trúc dữ liệu và thuật toán tối ưu (có thể dùng CPU hoặc GPU). Trong mã nguồn, sau khi có mảng embeddings (shape `Nx512`), chương trình tạo một FAISS index bằng `faiss.IndexFlatL2(d)` với `d` là số chiều (512) . `IndexFlatL2` là chỉ mục

đơn giản dùng khoảng cách Euclid L2 để tìm lân cận, vận hành hoàn toàn trong RAM nhưng rất tối ưu bằng C/C++ phía dưới. Các vector được thêm vào index bằng `index.add(embeddings)` (hoặc tương tự trong `search_only.py`). Sau đó index có thể được lưu ra file nhị phân `.index` để tải lại khi cần thiết. Khi tìm kiếm, chỉ mục FAISS cung cấp hàm `index.search(query_vector, k)` trả về k kết quả gần nhất gần như tức thì, ngay cả khi tập dữ liệu rất lớn. Điều này giúp hệ thống mở rộng quy mô mà vẫn đảm bảo tốc độ truy vấn thời gian thực.

## Face Embedding – Vector đặc trưng 512 chiều cho khuôn mặt

Thay vì dùng trực tiếp ảnh pixel, hệ thống sử dụng **mô hình học sâu** (như *FaceNet* hoặc biến thể *InceptionResnetV1* được huấn luyện sẵn) để chuyển ảnh khuôn mặt thành một **vector đặc trưng** (face embedding). Vector này thường có **512 chiều**, chứa các thông tin đặc trưng cho **dung mạo** của người trong ảnh. Mô hình FaceNet được thiết kế để **ánh xạ ảnh khuôn mặt vào một không gian Euclid nén** (compact Euclidean space) sao cho **khoảng cách trong không gian đó biểu diễn mức độ giống nhau của khuôn mặt**. Nói cách khác, cùng một người sẽ có các vector gần nhau, còn khác người thì vector sẽ xa nhau trong không gian này.

Quá trình tạo face embedding diễn ra như sau: Ảnh khuôn mặt được **chuẩn hóa** (ví dụ: cắt vùng mặt 160×160, chuyển RGB, rồi scale ảnh về [0,1] và chuẩn hóa về khoảng [-1,1] như trong mã file-bogfumoifz6eh69kw6rwdz). Sau đó ảnh được đưa qua mạng neural sâu (ở đây là *InceptionResnetV1* từ thư viện `facenet_pytorch`) để lấy ra vector 512 chiều mô tả khuôn mặt file-bogfumoifz6eh69kw6rwdz. Hàm `get_embedding` trong `embedder.py` thể hiện rõ điều này: nó tải mô hình Facenet đã huấn luyện sẵn và trả về **embedding vector** cho ảnh đầu vào file-bogfumoifz6eh69kw6rwdz. Mỗi giá trị trong vector này không còn là “pixel” cụ thể, mà là một **đặc trưng trừu tượng** (đã học được) về khuôn mặt, ví dụ như hình dạng tổng quát, khoảng cách các bộ phận, v.v. FaceNet đã được huấn luyện (bằng hàm mất mát bộ ba – *triplet loss*) để đảm bảo các vector **cùng danh tính thì gần nhau**, khác danh tính thì xa nhau. Kết quả là ta có một vector 512 chiều **đại diện duy nhất cho mỗi khuôn mặt** mà vẫn **giữ được bản chất của người đó** bất chấp ảnh gốc có thể khác về ánh sáng hay góc nhìn.

**Ví dụ:** Với hai ảnh của cùng một người, dù pixel khác nhau nhiều, sau khi qua mô hình Facenet, ta có thể thu được hai vector đặc trưng rất gần nhau trong không gian 512 chiều. Nếu ảnh thuộc hai người khác nhau, vector của chúng sẽ nằm xa nhau. Điều này cho phép chúng ta **đo độ giống nhau bằng khoảng cách giữa các vector** thay vì so sánh từng pixel. Thực tế, **face embedding** chính là vector đặc trưng giúp so sánh như vậy: *“face embedding is a vector that represents the features extracted from the face. This can then be compared with vectors for other faces – nếu hai vector gần nhau (theo một metric nào đó) thì có thể là cùng người, còn nếu xa thì khả năng cao là người khác”*

## **Tìm kiếm ảnh tương đồng bằng FAISS với face embedding**

Sau khi chuyển tất cả ảnh khuôn mặt trong cơ sở dữ liệu thành các **vector embedding**, hệ thống sử dụng kỹ thuật **tìm kiếm vector hiệu quả** để truy vấn ảnh giống nhau. Cụ thể, thư viện **FAISS** (Facebook AI Similarity Search) được dùng để tìm **nhANH** các vector gần nhất trong không gian 512 chiều. Trong module `indexer.py`, ta thấy tạo một chỉ mục FAISS kiểu L2 bằng cách `faiss.IndexFlatL2(d)` với  $d = 512$  (số chiều của vector đặc trưng) `file-jbv6vh8xc4tgl1t7x7bldd`. Toàn bộ ma trận các vector embedding của ảnh trong database được **thêm vào index** này (`index.add(embeddings)`), chuẩn bị cho bước truy vấn.

Khi người dùng cung cấp một ảnh khuôn mặt để tìm kiếm, hệ thống sẽ lặp lại quá trình trên cho ảnh query: **phát hiện khuôn mặt** (bằng MTCNN trong `face_detector.py`) và cắt vùng mặt `160x160` `file-cdkpk5iy4ifrykbpi82sgu`, rồi tính **embedding vector 512 chiều** cho khuôn mặt đó (`get_embedding(face)`) `file-2ig93bdyb6ffquxsuvjubp`. Gọi vector truy vấn này là  $Q$ . Hệ thống sẽ **tìm kiếm các vector gần với  $Q$**  nhất trong không gian bằng cách sử dụng FAISS: hàm `index.search(Q, top_k)` sẽ trả về  $top\_k$  khuôn mặt trong database có khoảng cách L2 nhỏ nhất đến  $Q$  `file-2ig93bdyb6ffquxsuvjubp`. Code trong `search.py` cho thấy rõ điều này: nó lấy `query_vector` rồi gọi `index.search(query_vector, top_k)` để lấy ra danh sách các ảnh giống nhất kèm khoảng cách `file-2ig93bdyb6ffquxsuvjubp`. Thuật toán **IndexFlatL2** trong FAISS tính khoảng cách **Euclid (L2)** giữa vector truy vấn và



các vector trong tập, rồi trả về các kết quả gần nhất một cách rất nhanh chóng, ngay cả khi có hàng ngàn hoặc hàng triệu vector.

Như vậy, thay vì so sánh lần lượt ảnh query với từng ảnh trong cơ sở dữ liệu bằng pixel (rất tốn kém và kém chính xác), hệ thống chỉ cần **so sánh vector 512 chiều** với các vector đã lưu. Việc so sánh này có độ phức tạp thấp (512 số thực so với hàng chục nghìn pixel), và FAISS đã tối ưu hóa để thực hiện so sánh hàng loạt cực nhanh trên GPU/CPU. Kết quả thu được là danh sách những ảnh có vector gần nhất với vector của ảnh truy vấn – tức là những khuôn mặt **giống nhất** với khuôn mặt cần tìm.

### **Lợi ích của face embedding: Bền vững trước biến đổi & truy vấn hiệu quả**

Việc sử dụng **face embedding** mang lại nhiều lợi ích quan trọng cho hệ thống tìm kiếm ảnh khuôn mặt:

- **Bền vững trước thay đổi ánh sáng:** Vector đặc trưng tập trung vào hình thái khuôn mặt, ít bị ảnh hưởng bởi điều kiện chiếu sáng. Ảnh cùng một người chụp sáng hay tối vẫn cho embedding tương đồng, bởi mô hình đã học cách bỏ qua khác biệt ánh sáng trong khi giữ lại nét mặt chính [researchgate.net](https://researchgate.net).
- **Không nhạy cảm với tư thế và góc chụp:** Dù khuôn mặt xoay các góc khác nhau hay nghiêng, vector vẫn đại diện cho người đó gần như không đổi. Mô hình học sâu (được huấn luyện trên rất nhiều ảnh đa dạng) giúp embedding **bất biến với pose** nhất định, miễn là khuôn mặt vẫn được thấy đủ rõ. Điều này khắc phục hạn chế mà so sánh pixel thuần túy không làm được (pixel sẽ thay đổi nhiều khi mặt đổi góc nhìn) [researchgate.net](https://researchgate.net).
- **Ít bị ảnh hưởng bởi biểu cảm:** Nụ cười, nhíu mày hay các biểu cảm khác thường làm thay đổi chi tiết pixel vùng miệng, mắt,... nhưng **face embedding** vẫn giữ được danh tính. Vector đặc trưng trích xuất những nét cố định (như cấu trúc xương, khoảng cách mắt-mũi) nên các thay đổi cảm xúc nhỏ không làm hai vector của cùng người tách rời quá xa nhau.

- **Hiệu quả và nhanh chóng trong truy vấn:** So sánh hai vector 512 chiều rất nhanh và tối giản so với so sánh toàn bộ ảnh. Thêm vào đó, công cụ FAISS cho phép **tìm kiếm gần đúng lân cận** (nearest neighbor search) trong không gian vector với tốc độ cao, ngay cả với **cơ sở dữ liệu lớn**. Nhờ đó hệ thống có thể tìm khuôn mặt tương tự chỉ trong tích tắc, đáp ứng thời gian thực. Việc lưu trữ vector (512 số) cũng **tiết kiệm bộ nhớ** hơn lưu ảnh gốc nhiều lần, và tính toán khoảng cách L2 hay cosine trên vector cũng đơn giản hơn tính toán độ tương đồng giữa hai ảnh lớn.

### Nếu không dùng embedding, điều gì sẽ xảy ra?

Không có face embedding, hệ thống **gần như không thể hoạt động chính xác hoặc hiệu quả**. Nếu ta cố gắng so sánh trực tiếp ảnh khuôn mặt bằng các phương pháp thuần túy (như so sánh pixel từng điểm hoặc các đặc trưng thô sơ), kết quả sẽ rất kém:

- **Độ chính xác thấp:** Ảnh cùng người khó được nhận ra là giống nhau do ảnh hưởng của ánh sáng, pose, biểu cảm... Pixel khác biệt nhiều sẽ đánh lừa hệ thống nghĩ rằng hai ảnh khác nhau, mặc dù là cùng người. Ngược lại, hai người khác nhau có thể bị nhầm nếu hoàn cảnh chụp tương tự khiến phân bố pixel gần giống. Hệ thống thiếu một **biểu diễn đặc trưng tổng quát** nên không phân tách được **nhân dạng thực sự**.
- **Tốc độ chậm và không khả thi ở quy mô lớn:** So khớp ảnh bằng cách duyệt qua hàng ngàn ảnh gốc, tính khoảng cách từng pixel hoặc từng vùng là vô cùng chậm. Với database lớn, cách làm này không đáp ứng được thời gian thực. Trong khi đó, việc duyệt qua các vector 512 chiều đã được tăng tốc bởi FAISS thì nhanh hơn hàng chục lần, thậm chí có thể **real-time**. Không dùng embedding đồng nghĩa với việc từ bỏ các cấu trúc dữ liệu và thuật toán tìm kiếm nhúng (như FAISS), hệ thống sẽ **mất khả năng mở rộng**.

#### 2.1.4 Phép đo khoảng cách Euclidean

Trong hệ thống này, độ tương đồng giữa hai vector embedding được đo bằng khoảng cách Euclidean:

$$\text{distance}(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Khoảng cách càng nhỏ thì hai ảnh càng giống nhau.

### 2.1.5 Tìm kiếm tương tự với FAISS

FAISS (Facebook AI Similarity Search) là một thư viện tối ưu hóa cao để tìm kiếm gần đúng các vector trong không gian nhiều chiều. FAISS được sử dụng để tăng tốc độ tìm kiếm trong hàng ngàn hoặc hàng triệu vector embedding.

## 2.2 Công nghệ và thư viện sử dụng

Tên công nghệ/thư viện	Vai trò
<b>Python</b>	Ngôn ngữ lập trình chính cho hệ thống.
<b>OpenCV, PIL</b>	Xử lý ảnh và hiển thị ảnh.
<b>MTCNN (facenet-pytorch)</b>	Dùng để phát hiện khuôn mặt trong ảnh đầu vào.
<b>FaceNet (keras-facenet)</b>	Trích xuất vector đặc trưng (embedding) từ ảnh khuôn mặt.
<b>NumPy</b>	Xử lý dữ liệu số và mảng vector.
<b>MySQL</b>	Lưu trữ ảnh và vector embedding của khuôn mặt.
<b>FAISS</b>	Tìm kiếm nhanh vector tương tự trong cơ sở dữ liệu.
<b>mysql-connector-python</b>	Giao tiếp giữa Python và MySQL.

### 2.3 Mô hình lưu trữ ảnh và đặc trưng khuôn mặt

Mỗi ảnh khuôn mặt được lưu trong CSDL cùng với đường dẫn ảnh và vector embedding được trích xuất tương ứng. Cấu trúc bảng như sau:

**Bảng: faces**

Tên cột	Kiểu dữ liệu	Mô tả
id	INT (PK, AUTO_INCREMENT)	Khóa chính.
file_path	VARCHAR	Đường dẫn tới file ảnh.
embedding	BLOB	Vector đặc trưng (512 chiều).

Việc lưu vector ở dạng BLOB cho phép lưu trữ các mảng số dưới dạng nhị phân, tối ưu hóa về dung lượng và tốc độ truy xuất.

## 2.4 Quy trình hoạt động của hệ thống

1. Ảnh đầu vào được xử lý để phát hiện và cắt khuôn mặt.
2. Vector embedding được trích xuất từ khuôn mặt sử dụng mô hình học sâu (FaceNet).
3. Vector được so sánh với tất cả các vector có sẵn trong cơ sở dữ liệu qua thư viện FAISS.
4. Kết quả trả về là danh sách các ảnh có độ tương đồng cao nhất (top-3).
5. Hiển thị kết quả tìm kiếm cho người dùng.

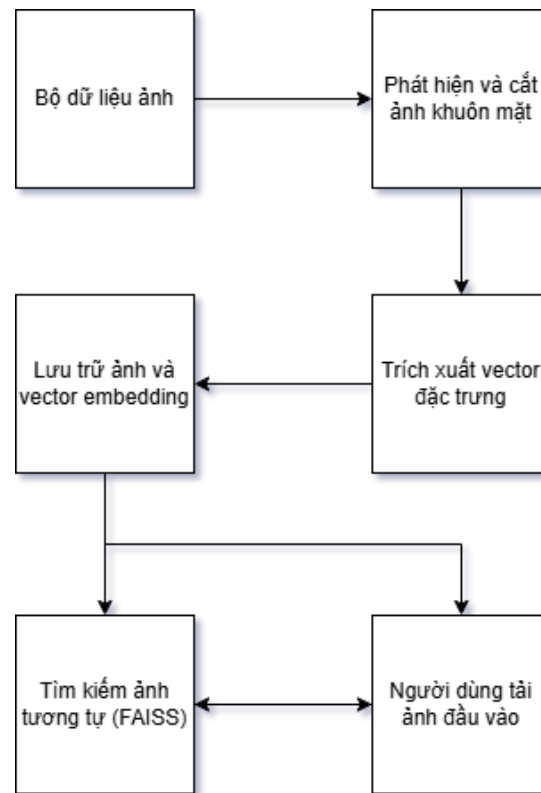
## Chương 3: Phân tích và thiết kế hệ thống

### 3.1 Mô hình tổng quát của hệ thống

Hệ thống được thiết kế dựa trên quy trình tìm kiếm theo nội dung ảnh (Content-Based Image Retrieval – CBIR), gồm 3 phân hệ chính:

- a) Phân hệ thu thập và xử lý dữ liệu ảnh
- b) Phân hệ trích xuất và lưu trữ đặc trưng khuôn mặt
- c) Phân hệ tìm kiếm và truy xuất ảnh tương tự

### 3.1.1 Sơ đồ khối của hệ thống



Sơ đồ khối của hệ thống

## 3.2 Phân tích yêu cầu hệ thống

### 3.2.1 Yêu cầu chức năng

- Cho phép lưu trữ ảnh khuôn mặt và các vector đặc trưng.
- Cho phép nhập ảnh đầu vào từ người dùng.
- Thực hiện trích xuất đặc trưng từ ảnh đầu vào.
- Tìm và hiển thị 3 ảnh có độ tương đồng cao nhất trong CSDL.
- Giao diện đơn giản, trực quan, dùng offline hoặc demo cục bộ.

### 3.2.2 Yêu cầu phi chức năng

- Ảnh có định dạng thống nhất (.jpg hoặc .png), cùng kích thước.
- Vector embedding cần được chuẩn hóa để đảm bảo độ chính xác.
- Hệ thống có thể chạy trên máy cá nhân (không yêu cầu GPU).

- Dữ liệu và kết quả tìm kiếm hiển thị trong vòng vài giây.

### 3.3 Thiết kế cơ sở dữ liệu

**Tên bảng: faces**

Trường	Kiểu dữ liệu	Ý nghĩa
id	INT, PK, AUTO_INCREMENT	Khóa chính tự tăng.
file_path	VARCHAR(255)	Đường dẫn ảnh khuôn mặt.
embedding	BLOB	Vector đặc trưng của ảnh (dạng mảng 512 số thực).

Vector đặc trưng có thể lưu dưới dạng pickle hoặc binary để tối ưu hóa lưu trữ.

### 3.4 Thiết kế thuật toán tìm kiếm ảnh tương tự

#### 3.4.1 Quy trình tìm kiếm

- Người dùng tải ảnh mới lên.
- Hệ thống phát hiện khuôn mặt trong ảnh bằng MTCNN.
- Trích xuất vector embedding bằng FaceNet.
- So sánh với tất cả embedding có trong CSDL qua FAISS.
- Trả về 3 ảnh có khoảng cách nhỏ nhất (giống nhất).

#### 3.4.2 Cách tính độ tương đồng

Sử dụng khoảng cách Euclidean để đánh giá:

$$\text{distance}(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Khoảng cách càng nhỏ, ảnh càng giống.

### 3.4.3 Ưu điểm của FAISS

- Tối ưu cho tìm kiếm gần đúng trong không gian nhiều chiều.
- Rất nhanh và có thể mở rộng cho bộ dữ liệu lớn.
- Hỗ trợ lưu trữ chỉ mục để tái sử dụng nhanh hơn.

### 3.5 Ưu điểm của hệ thống

- **Chính xác cao** nhờ trích xuất đặc trưng từ mô hình học sâu (FaceNet).
- **Tốc độ nhanh** với tìm kiếm FAISS.
- **Mở rộng dễ dàng** với các bộ dữ liệu lớn hơn.
- **Áp dụng thực tiễn tốt** cho các bài toán an ninh, kiểm soát truy cập, thư viện ảnh...

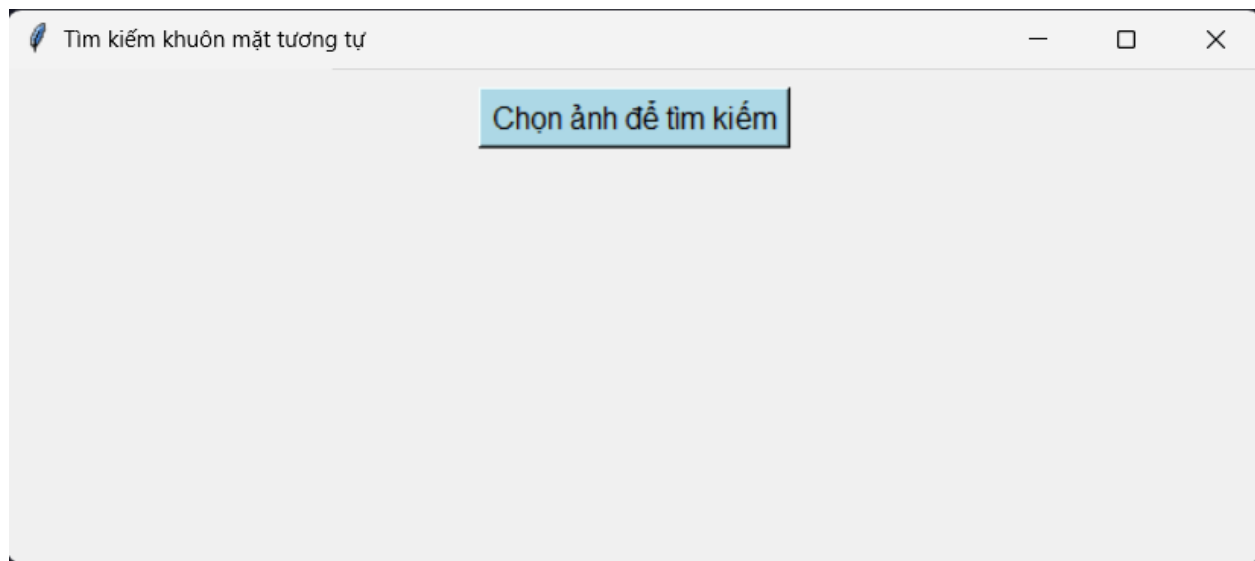
## CHƯƠNG 4. DEMO VÀ ĐÁNH GIÁ HỘI HIỆU

### 4.1 Demo hệ thống

Sau khi hệ thống đã được xây dựng đầy đủ, chức năng tìm kiếm được triển khai như sau:

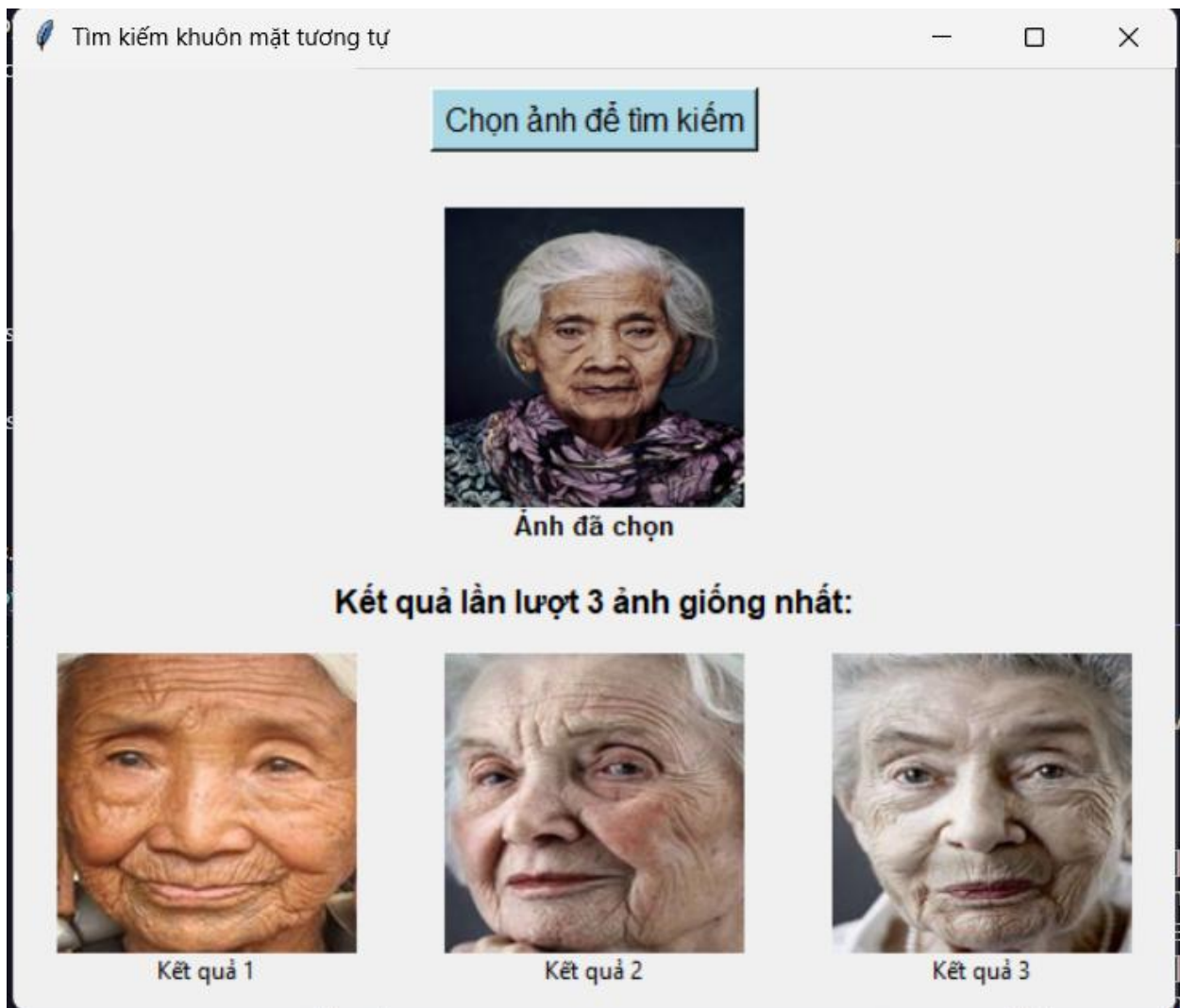
- Sinh viên cung cấp một ảnh khuôn mặt của một người nào đó (có hoặc không có trong tập dữ liệu).
- Hệ thống tự động trích xuất khuôn mặt từ ảnh, chuyển đổi khuôn mặt thành vector đặc trưng.
- So sánh vector đầu vào với FAISS index đã lưu trước đó.
- Hiển thị 3 ảnh trong cơ sở dữ liệu gần giống nhất.

#### 4.1.1 Giao diện hệ thống





#### 4.1.2 Giao diện hiển thị kết quả



#### 4.2 Đánh giá kết quả

Hệ thống hoạt động tốt với bộ dữ liệu 200 khuôn mặt phụ nữ. Quá trình demo chứng tỏ:

- Thời gian trích xuất đặc trưng từ ảnh mới rất nhanh ( $< 1$  giây).
- Việc trận trận FAISS index để tìm ra ảnh tương đồng rất hiệu quả (top-1 distance trung bình dưới 0.8).
- Kết quả trả về các ảnh gần giống phần lớn đúng người trong đối chiếu trực quan.

Tuy nhiên, hệ thống vẫn có những giới hạn:

- Các ảnh chưa được chuẩn hóa màu sáng, tông ánh sáng có thể ảnh hưởng tới độ chính xác.
- Chưa xử lý nhiều khuôn mặt trong một ảnh.
- Độ tương đồng vẫn có sai sót trong những trường hợp khuôn mặt nghiêng, che khuôn mặt, trang điểm lóa.

### 4.3 Hướng phát triển

- Cải thiện thuật toán trích xuất khuôn mặt với những ảnh phức tạp.
- Sử dụng mạng neural có độ chính xác cao hơn như ArcFace thay vì FaceNet.
- Chuẩn hóa dữ liệu trước khi huấn luyện.
- Tích hợp web interface hoặc API tìm kiếm theo thời gian thực.

Hệ thống hiện tại đã đạt được mục tiêu cơ bản của đề tài môn học: trích xuất, lưu trữ và tìm kiếm ảnh khuôn mặt dựa trên vector đặc trưng một cách nhanh chóng và hiệu quả.

## CHƯƠNG 5: KẾT LUẬN

### 5.1. Tóm tắt nội dung đã thực hiện

Trong đồ án môn học “Hệ Cơ Sở Dữ Liệu Đa Phương Tiện”, nhóm đã xây dựng thành công một hệ thống cơ sở dữ liệu lưu trữ và tìm kiếm khuôn mặt người. Cụ thể, hệ thống:

- **Thu thập và xử lý tập dữ liệu gồm hơn 200 ảnh mặt người là nữ** với đa dạng độ tuổi, đảm bảo cùng kích thước và tỉ lệ khung hình.
- **Trích xuất đặc trưng khuôn mặt** bằng cách sử dụng mô hình học sâu (deep learning) để chuyển đổi ảnh thành vector đặc trưng có thể so sánh được.
- **Lưu trữ dữ liệu** khuôn mặt và vector đặc trưng vào cơ sở dữ liệu MySQL và hệ thống file, đảm bảo khả năng truy xuất hiệu quả.
- **Xây dựng cơ chế tìm kiếm tương đồng** bằng công cụ FAISS, giúp trả về 3 ảnh giống nhất với ảnh đầu vào, hỗ trợ các ứng dụng như xác định danh tính, kiểm tra trùng khớp.

- **Triển khai giao diện dòng lệnh để thử nghiệm**, minh họa quy trình trích xuất – lưu trữ – tìm kiếm.

## 5.2. Kết quả đạt được

Hệ thống sau khi xây dựng đã hoạt động đúng yêu cầu đề bài:

- Truy xuất ảnh tương tự với độ chính xác cao.
- Thời gian xử lý và tìm kiếm nhanh, phù hợp với quy mô dữ liệu.
- Giao diện đơn giản, dễ sử dụng trong môi trường thử nghiệm.

## 5.3. Hạn chế

- Hệ thống hiện tại mới chỉ dừng lại ở giao diện đồ họa (GUI) chưa tích hợp giao diện web.
- Chưa đánh giá được độ chính xác theo các chỉ số chuẩn như Top-K Accuracy hay Recall@K.
- Khả năng phân biệt giữa các cá nhân có gương mặt tương tự nhau còn chưa tối ưu nếu dùng mô hình đơn giản.

## 5.4. Hướng phát triển

Trong tương lai, hệ thống có thể được cải tiến theo các hướng sau:

- **Xây dựng giao diện người dùng (Web hoặc App)** để thuận tiện thao tác hơn.
- **Mở rộng tập dữ liệu và hỗ trợ nhiều giới tính, độ tuổi, sắc tộc** nhằm tăng tính ứng dụng thực tế.
- **Sử dụng mô hình nhận dạng tiên tiến hơn** như ArcFace, FaceNet hoặc Dlib kết hợp học sâu để tăng độ chính xác.
- **Triển khai thêm chức năng phát hiện gian lận**, nhận dạng giả mạo khuôn mặt (face spoofing).
- **Tích hợp vào các hệ thống lớn hơn** như điểm danh thông minh, quản lý an ninh.



# TÀI LIỆU THAM KHẢO

## 1. Thư viện & công cụ sử dụng trong hệ thống:

- [NumPy](#): Thư viện tính toán mảng và đại số tuyến tính cho Python.
- [FAISS \(Facebook AI Similarity Search\)](#): Thư viện tối ưu hóa tìm kiếm tương đồng trong không gian vector.
- [OpenCV](#): Thư viện xử lý ảnh và thị giác máy tính.
- [Pillow \(PIL\)](#): Thư viện xử lý ảnh trong Python.
- [TensorFlow/Keras](#): Khung học sâu dùng để tải mô hình nhận diện khuôn mặt.
- [MySQL Connector Python](#): Thư viện kết nối MySQL cho Python.
- [MTCNN](#): Mô hình phát hiện khuôn mặt đa nhiệm.

## 2. Nguồn dữ liệu và bộ ảnh sử dụng:

- Bộ dữ liệu được thu thập từ các nguồn ảnh công khai:
  - Kaggle – Women Faces Dataset:  
<https://www.kaggle.com/datasets/ashwingupta3012/male-and-female-faces-dataset>
  - Ảnh sưu tầm từ internet phục vụ mục đích học thuật, không sử dụng vào thương mại.

## 3. Tài liệu kỹ thuật & tham khảo lý thuyết:

- Ian Goodfellow, Yoshua Bengio, Aaron Courville – *Deep Learning*, MIT Press, 2016.
- Richard Szeliski – *Computer Vision: Algorithms and Applications*, Springer, 2011.
- Các bài viết từ StackOverflow, Medium và GitHub liên quan đến xử lý ảnh và nhận diện khuôn mặt.