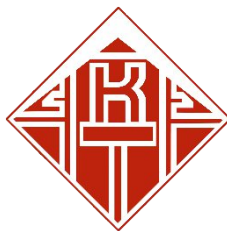


TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG  
KHOA CÔNG NGHỆ THÔNG TIN



**ĐỒ ÁN TỐT NGHIỆP**  
**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**ĐỀ TÀI:**

**XÂY DỰNG WEBSITE ĐIỂM DANH SINH VIÊN  
BẰNG NHẬN DIỆN KHUÔN MẶT**

Người hướng dẫn: **TS. ĐOÀN DUY BÌNH**  
Sinh viên thực hiện: **BÙI NGỌC KHẮC LÂM**  
Mã số sinh viên: **2151220262**  
Lớp: **21CT5**

**Đà Nẵng, 12/2025**

**TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN TỐT NGHIỆP**  
**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**ĐỀ TÀI:**

**XÂY DỰNG WEBSITE ĐIỂM DANH SINH VIÊN**  
**BẰNG NHẬN DIỆN KHUÔN MẶT**

Người hướng dẫn: **TS. ĐOÀN DUY BÌNH**  
Sinh viên thực hiện: **BÙI NGỌC KHẮC LÂM**  
Mã số sinh viên: **2151220262**  
Lớp: **21CT5**

**Đà Nẵng, 12/2025**

## TÓM TẮT

Tên đề tài: Xây dựng website điểm danh sinh viên bằng nhận diện khuôn mặt

Sinh viên thực hiện: Bùi Ngọc Khắc Lâm

Mã số SV: 2151220262      Lớp: 21CT5

Đề tài “Xây dựng website điểm danh sinh viên bằng nhận diện khuôn mặt” được thực hiện nhằm ứng dụng công nghệ trí tuệ nhân tạo vào công tác quản lý điểm danh trong các cơ sở giáo dục. Hệ thống giúp tự động hóa quá trình điểm danh, thay thế phương pháp thủ công truyền thống vốn tốn nhiều thời gian và dễ xảy ra sai sót. Website cho phép người dùng tạo lớp học, quản lý danh sách sinh viên và tiến hành điểm danh thông qua camera. Hệ thống sử dụng công nghệ nhận diện khuôn mặt để xác định danh tính sinh viên dựa trên hình ảnh thu được, sau đó tự động ghi nhận kết quả điểm danh. Dữ liệu điểm danh được lưu trữ và hiển thị dưới dạng báo cáo giúp việc theo dõi trở nên dễ dàng và chính xác hơn.

Về mặt kỹ thuật, website được phát triển với Python và thư viện OpenCV để xử lý hình ảnh và nhận diện khuôn mặt. Phần giao diện người dùng được xây dựng bằng HTML, CSS, JavaScript, đảm bảo thân thiện và dễ sử dụng. Cơ sở dữ liệu được thiết kế nhằm lưu trữ thông tin sinh viên, lớp học và lịch sử điểm danh một cách an toàn, hiệu quả. Hệ thống góp phần nâng cao tính chính xác, tiết kiệm thời gian và hiện đại hóa công tác quản lý điểm danh trong môi trường giáo dục. Đây là bước ứng dụng thực tiễn của trí tuệ nhân tạo vào lĩnh vực quản lý, đồng thời mở ra hướng phát triển cho các hệ thống tự động trong tương lai.

## ***LỜI NÓI ĐẦU***

Trong quá trình học tập và thực hiện đề tài “Xây dựng website điểm danh sinh viên bằng nhận diện khuôn mặt”, em đã nhận được sự giúp đỡ, chỉ bảo tận tình của quý thầy cô và sự hỗ trợ từ bạn bè. Đây là nguồn động viên to lớn giúp em hoàn thành tốt bài đồ án này. Trước hết, em xin bày tỏ lòng biết ơn sâu sắc đến thầy TS. Đoàn Duy Bình, người đã tận tình chỉ dẫn, định hướng và hỗ trợ em trong suốt quá trình nghiên cứu, từ khâu lên ý tưởng đến khi hoàn thiện đề tài.

Em cũng xin gửi lời cảm ơn đến quý thầy cô Khoa Công nghệ thông tin, Trường Đại học Kiến trúc Đà Nẵng, những người đã truyền đạt cho em nhiều kiến thức quý báu trong suốt thời gian học tập. Những kiến thức đó là nền tảng quan trọng để em có thể vận dụng vào việc thực hiện đồ án tốt nghiệp này.

Cuối cùng, em xin cảm ơn gia đình và bạn bè đã luôn động viên, khích lệ, giúp đỡ em trong suốt quá trình học tập và hoàn thành đề tài.

Em xin chân thành cảm ơn!

## ***CAM ĐOAN***

Em xin cam đoan rằng đề tài “Xây dựng website điểm danh sinh viên bằng nhận diện khuôn mặt” là kết quả nghiên cứu và thực hiện của riêng em, dưới sự hướng dẫn của thầy TS. Đoàn Duy Bình.

Các nội dung, số liệu, hình ảnh và kết quả trình bày trong đồ án được em thu thập, phân tích và xây dựng một cách trung thực. Những tài liệu, nguồn tham khảo được sử dụng trong quá trình thực hiện đều được trích dẫn rõ ràng và đầy đủ theo quy định.

Em hoàn toàn chịu trách nhiệm về tính trung thực và liêm chính học thuật của đồ án này. Nếu phát hiện có sự sao chép hoặc gian lận, em xin chịu hoàn toàn trách nhiệm trước nhà trường và quy định hiện hành.

Sinh viên thực hiện

{Chữ ký, họ và tên sinh viên}

# MỤC LỤC

|                                                                              |    |
|------------------------------------------------------------------------------|----|
| LỜI NÓI ĐẦU .....                                                            | i  |
| CAM ĐOAN .....                                                               | ii |
| DANH SÁCH CÁC HÌNH VẼ, BẢNG .....                                            | v  |
| DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT .....                                    | vi |
| MỞ ĐẦU .....                                                                 | 1  |
| 1. Tổng quan tình hình nghiên cứu đề tài .....                               | 1  |
| 2. Lý do chọn đề tài .....                                                   | 1  |
| 3. Mục tiêu đề tài .....                                                     | 1  |
| 4. Phạm vi và đối tượng nghiên cứu .....                                     | 1  |
| 5. Phương pháp nghiên cứu .....                                              | 1  |
| Chương 1: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ LIÊN QUAN .....                       | 3  |
| 1.1. Các khái niệm cơ bản về thị giác máy tính và trí tuệ nhân tạo .....     | 3  |
| 1.1.1. Trí tuệ nhân tạo ( <i>Artificial Intelligence – AI</i> ) .....        | 3  |
| 1.1.2. Thị giác máy tính ( <i>Computer Vision</i> ) .....                    | 4  |
| 1.2. Công nghệ nhận diện khuôn mặt .....                                     | 5  |
| 1.2.1. Lịch sử phát triển .....                                              | 5  |
| 1.2.2. Nhận diện khuôn mặt là gì .....                                       | 5  |
| 1.2.3. Công nghệ nhận diện khuôn mặt hoạt động như thế nào? .....            | 6  |
| 1.2.4. Ứng dụng của công nghệ nhận diện khuôn mặt trong các ngành nghề ..... | 7  |
| 1.2.5. Ưu và nhược điểm của nhận diện khuôn mặt .....                        | 7  |
| 1.3. Các thư viện và công cụ hỗ trợ .....                                    | 8  |
| 1.3.1. Các thư viện chính .....                                              | 8  |
| 1.3.2. Các công cụ hỗ trợ nhận dạng khuôn mặt .....                          | 12 |
| 1.4. Công nghệ phát triển web .....                                          | 18 |
| 1.4.1. Python .....                                                          | 18 |
| 1.4.2. Flask .....                                                           | 19 |
| 1.4.3. MySQL .....                                                           | 20 |
| 1.5. Kiến trúc hệ thống web ứng dụng trí tuệ nhân tạo .....                  | 21 |
| Chương 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG .....                               | 23 |
| 2.1. Khảo sát yêu cầu thực tế .....                                          | 23 |
| 2.1.1. Thực trạng công tác điểm danh hiện nay .....                          | 23 |
| 2.1.2. Mục tiêu khảo sát .....                                               | 23 |
| 2.1.3. Đối tượng khảo sát .....                                              | 23 |

|                    |                                                               |    |
|--------------------|---------------------------------------------------------------|----|
| 2.1.4.             | <i>Kết quả khảo sát</i>                                       | 23 |
| 2.1.5.             | <i>Đề xuất hướng giải pháp</i>                                | 24 |
| 2.2.               | Phân tích hệ thống                                            | 24 |
| 2.2.1.             | <i>Mô hình UseCase</i>                                        | 24 |
| 2.2.2.             | <i>Sơ đồ chức năng</i>                                        | 26 |
| 2.2.3.             | <i>Biểu đồ tuần tự</i>                                        | 28 |
| 2.2.4.             | <i>Thiết kế cơ sở dữ liệu</i>                                 | 31 |
| 2.2.5.             | <i>Giao diện người dùng</i>                                   | 32 |
| Chương 3:          | XÂY DỰNG VÀ TRIỂN KHAI HỆ THỐNG                               | 44 |
| 3.1.               | Môi trường và công cụ phát triển                              | 44 |
| 3.1.1.             | <i>Công cụ phát triển</i>                                     | 44 |
| 3.1.2.             | <i>Quy trình triển khai hệ thống</i>                          | 44 |
| 3.2.               | Xây dựng backend và API quản lý lớp học, sinh viên, điểm danh | 45 |
| 3.2.1.             | <i>Mục tiêu</i>                                               | 45 |
| 3.2.2.             | <i>Một số đoạn mã minh họa</i>                                | 46 |
| 3.3.               | Xây dựng frontend - giao diện người dùng                      | 52 |
| 3.4.               | Tích hợp mô hình nhận diện khuôn mặt vào hệ thống             | 53 |
| 3.5.               | Triển khai hệ thống trên môi trường thực nghiệm               | 54 |
| KẾT LUẬN           |                                                               | 57 |
| 1.                 | Kết luận chung                                                | 57 |
| 2.                 | Kết quả đạt được                                              | 57 |
| 3.                 | Đóng góp của đề tài                                           | 57 |
| 4.                 | Đề xuất và kiến nghị                                          | 57 |
| TÀI LIỆU THAM KHẢO |                                                               | 58 |
| Tiếng Việt         |                                                               | 58 |
| Tiếng Anh          |                                                               | 58 |

## DANH SÁCH CÁC HÌNH VẼ

|                                                                            |    |
|----------------------------------------------------------------------------|----|
| Hình 1: Công nghệ nhận diện khuôn mặt.....                                 | 6  |
| Hình 2: Khái niệm công cụ Git.....                                         | 15 |
| Hình 3: Khái niệm về Flask.....                                            | 19 |
| Hình 4: Kiến trúc hệ thống Website.....                                    | 22 |
| Hình 5: Mô hình UseCase tổng quát.....                                     | 24 |
| Hình 6: Sơ đồ chức năng.....                                               | 26 |
| Hình 7: Sơ đồ hoạt động quy trình điểm danh.....                           | 27 |
| Hình 8: Sơ đồ hoạt động Trainning sinh viên.....                           | 27 |
| Hình 9: Biểu đồ tuần tự quản lý sinh viên.....                             | 28 |
| Hình 10: Biểu đồ tuần tự quản lý lớp học.....                              | 29 |
| Hình 11: Biểu đồ tuần tự Training model.....                               | 30 |
| Hình 12: Biểu đồ tuần tự điểm danh.....                                    | 30 |
| Hình 13: Mô hình quan hệ dữ liệu của hệ thống.....                         | 31 |
| Hình 14: Giao diện trang đăng nhập.....                                    | 37 |
| Hình 15: Giao diện trang chủ.....                                          | 38 |
| Hình 16: Giao diện quản lý sinh viên.....                                  | 39 |
| Hình 17: Giao diện quản lý người dùng.....                                 | 39 |
| Hình 18: Giao diện quản lý lớp học.....                                    | 40 |
| Hình 19: Giao diện điểm danh sinh viên trên hệ thống quản trị viên.....    | 41 |
| Hình 20: Giao diện điểm danh sinh viên công khai.....                      | 41 |
| Hình 21: Giao diện lịch sử điểm danh.....                                  | 42 |
| Hình 22: Giao diện báo cáo thống kê.....                                   | 43 |
| Hình 23: API lấy danh sách lớp học.....                                    | 46 |
| Hình 24: API tạo mới lớp học.....                                          | 47 |
| Hình 25: API tạo mới sinh viên.....                                        | 48 |
| Hình 26: API lấy danh sách sinh viên.....                                  | 49 |
| Hình 27: API nhận diện khuôn mặt.....                                      | 50 |
| Hình 28: API lấy danh sách điểm danh.....                                  | 51 |
| Hình 29: Mã code Giao diện điểm danh.....                                  | 52 |
| Hình 30: Đoạn mã khởi tạo mô hình phát hiện khuôn mặt SCRFD.....           | 53 |
| Hình 31: Đoạn mã xử lý và chuẩn hóa ảnh khuôn mặt trước khi nhận diện..... | 53 |
| Hình 32: Đoạn mã xử lý và chuẩn hóa ảnh khuôn mặt trước khi nhận diện..... | 54 |
| Hình 33: Đoạn mã so sánh đặc trưng khuôn mặt với cơ sở dữ liệu.....        | 54 |
| Hình 34: Đoạn mã trả về kết quả.....                                       | 54 |

|                                                                                |    |
|--------------------------------------------------------------------------------|----|
| Hình 35: Tập dữ liệu khuôn mặt thu thập cho quá trình huấn luyện mô hình. .... | 55 |
|--------------------------------------------------------------------------------|----|

## **DANH SÁCH CÁC BẢNG**

|                                                            |    |
|------------------------------------------------------------|----|
| Bảng 1: Mô tả mô hình UseCase.....                         | 25 |
| Bảng 2: Mô tả chi tiết bảng attendance_admin_accounts..... | 31 |
| Bảng 3: Mô tả chi tiết bảng attendance_cameras.....        | 32 |
| Bảng 4: Mô tả chi tiết bảng attendance_classes.....        | 33 |
| Bảng 5: Mô tả chi tiết attendance_history.....             | 34 |
| Bảng 6: Mô tả chi tiết bảng attendance_menus.....          | 35 |
| Bảng 7: Mô tả chi tiết bảng students.....                  | 36 |

## DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

| Từ viết tắt | Giải thích                                                                                                     |
|-------------|----------------------------------------------------------------------------------------------------------------|
| AI          | Artificial Intelligence - Trí tuệ nhân tạo                                                                     |
| API         | Application Programming Interface - Giao diện lập trình ứng dụng                                               |
| CNN         | Convolutional Neural Networks - Mạng nơ-ron tích chập                                                          |
| CRUD        | Create, Read, Update, Delete (Thêm, Xem, Sửa, Xóa)                                                             |
| SCRFD       | Sparsely Connected Regression for Face Detection - Một mô hình phát hiện khuôn mặt được sử dụng trong hệ thống |
| SCM         | Source Code Management - Quản lý mã nguồn                                                                      |
| HTTP        | Hypertext Transfer Protocol - Giao thức truyền tải siêu văn bản                                                |
| IP          | Internet Protocol - Giao thức mạng                                                                             |
| URL         | Uniform Resource Locator - Đường dẫn tham chiếu tài nguyên                                                     |

## **MỞ ĐẦU**

### **1. Tổng quan tình hình nghiên cứu đề tài**

Trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ trong lĩnh vực giáo dục, việc ứng dụng công nghệ vào quản lý và giảng dạy trở nên cần thiết hơn bao giờ hết. Một trong những hoạt động quan trọng tại các cơ sở giáo dục là công tác điểm danh sinh viên. Tuy nhiên, việc điểm danh thủ công truyền thống thường mất nhiều thời gian, dễ xảy ra sai sót và gian lận. Từ thực tế đó, đề tài “Xây dựng website điểm danh sinh viên bằng nhận diện khuôn mặt” được thực hiện nhằm mang đến một giải pháp hiện đại, tự động và hiệu quả hơn trong quản lý chuyên cần.

### **2. Lý do chọn đề tài**

Hiện nay việc nhận diện gương mặt đã trở nên cần thiết đối với nhiều ngành nghề, lĩnh vực khác nhau. Việc xây dựng một hệ thống gương mặt có thể đáp ứng được mọi nhu cầu sử dụng vẫn đang là một vấn đề nan giải mặt dù đã tồn tại rất nhiều mô hình nhận diện gương mặt. Để có thể giải bài toán này, ta cần phải thiết kế hệ thống có khả năng linh hoạt cao cũng như phạm vi ứng dụng lớn để người dùng có thể truy cập và sử dụng ở mọi lúc, mọi nơi.

### **3. Mục tiêu đề tài**

Xây dựng website có khả năng nhận diện và xác thực khuôn mặt sinh viên khi điểm danh, cung cấp chức năng quản lý lớp học, buổi học và dữ liệu điểm danh cho giảng viên, hỗ trợ quản trị viên quản lý tài khoản, dữ liệu sinh viên và giảng viên một cách hiệu quả.

### **4. Phạm vi và đối tượng nghiên cứu**

Phạm vi nghiên cứu của đề tài tập trung vào việc xây dựng hệ thống website điểm danh sinh viên bằng công nghệ nhận diện khuôn mặt, áp dụng trong môi trường trường học. Hệ thống hướng đến việc tự động hóa quá trình điểm danh, giúp nâng cao hiệu quả quản lý và giảm thiểu sai sót so với phương pháp thủ công.

Đối tượng nghiên cứu của đề tài là phương pháp và ứng dụng nhận diện sinh viên từ gương mặt. Trước mắt, chúng tôi sẽ thực hiện nhận diện những sinh viên thuộc khoa Công nghệ thông tin trường Đại học Kiến trúc Đà Nẵng.

### **5. Phương pháp nghiên cứu**

Đề tài được thực hiện dựa trên việc nghiên cứu tài liệu, phân tích yêu cầu thực tế và tham khảo các công nghệ liên quan đến nhận diện khuôn mặt. Phương pháp nghiên cứu

bao gồm khảo sát nhu cầu sử dụng trong nhà trường, thiết kế hệ thống theo mô hình web, và áp dụng công nghệ Python để xây dựng hệ thống. Ngoài ra, nhóm tiến hành kiểm thử và đánh giá kết quả nhằm đảm bảo tính chính xác và hiệu quả của ứng dụng.

## **Chương 1: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ LIÊN QUAN**

### **1.1. Các khái niệm cơ bản về thị giác máy tính và trí tuệ nhân tạo**

#### **1.1.1. Trí tuệ nhân tạo (Artificial Intelligence – AI)**

Trí tuệ nhân tạo (AI) là lĩnh vực của khoa học máy tính nghiên cứu và phát triển các hệ thống có khả năng mô phỏng trí tuệ con người như: học tập, suy luận, nhận thức và tự đưa ra quyết định. Mục tiêu của AI là giúp máy tính hiểu, phân tích và xử lý thông tin một cách thông minh, tương tự con người. [1]

Mục tiêu cốt lõi của AI là giúp máy tính hoặc máy móc có thể thực hiện các nhiệm vụ một cách tự động, thông minh và chính xác mà trước đây chỉ có con người mới làm được. Nhờ sự phát triển mạnh mẽ của phần cứng (GPU, TPU) và các thuật toán học sâu (Deep Learning), AI ngày nay đã đạt được nhiều thành tựu nổi bật và được ứng dụng rộng rãi trong hầu hết các lĩnh vực của đời sống hiện đại.

Trong thực tế, AI được sử dụng trong nhiều ứng dụng quen thuộc như: nhận diện khuôn mặt trên điện thoại, hệ thống gợi ý nội dung trên mạng xã hội, trợ lý ảo, xe tự hành, phân tích dữ liệu y khoa, phát hiện gian lận tài chính, xử lý ngôn ngữ tự nhiên (NLP), và đặc biệt là nhận dạng hình ảnh phục vụ cho các hệ thống điểm danh bằng khuôn mặt. Với khả năng phân tích và trích xuất đặc trưng khuôn mặt nhanh chóng, AI giúp nâng cao độ chính xác và tối ưu hóa quy trình điểm danh trong các trường học hoặc doanh nghiệp.

Các đặc điểm quan trọng của AI:

- Tính tự học (Self-learning): AI có khả năng cải thiện dần theo thời gian khi tiếp xúc nhiều hơn với dữ liệu. Với mỗi lần điểm danh, hệ thống có thể học thêm đặc trưng khuôn mặt và nâng cao độ chính xác.
- Tính tự thích nghi (Adaptability):
- Hệ thống AI có thể hoạt động trong nhiều môi trường ánh sáng, góc nhìn, chất lượng camera khác nhau — điều rất quan trọng đối với hệ thống điểm danh thực tế.
- Khả năng xử lý dữ liệu lớn (Big Data Processing):
- AI có thể phân tích lượng lớn hình ảnh hoặc video trong thời gian rất ngắn, giúp xử lý nhiều sinh viên cùng lúc.
- Tính dự đoán và khái quát hóa:
- AI không chỉ ghi nhớ dữ liệu đã học mà còn dự đoán chính xác ngay cả khi khuôn mặt có thay đổi nhỏ như tóc, kính, khẩu trang,...

Một số nhánh chính của AI bao gồm:

- **Machine Learning (Học máy):** Là phương pháp cho phép hệ thống học từ dữ liệu thông qua các thuật toán và mô hình thống kê. Thay vì lập trình thủ công từng bước, hệ thống tự tìm ra quy luật và cải thiện hiệu suất dựa trên việc phân tích dữ liệu đầu vào. Đây là nền tảng cho nhiều mô hình phân loại và nhận diện.
- **Deep Learning (Học sâu):** Là một nhánh của học máy sử dụng mạng nơ-ron nhân tạo nhiều tầng (Artificial Neural Networks). Deep Learning có khả năng xử lý dữ liệu lớn và phức tạp như hình ảnh, âm thanh, video. Trong nhận diện khuôn mặt, Deep Learning giúp mô hình học được các đặc trưng khuôn mặt sâu hơn và chính xác hơn so với các phương pháp truyền thống.
- **Computer Vision (Thị giác máy tính):** Là lĩnh vực giúp máy tính “nhìn thấy” và hiểu được nội dung trong hình ảnh hoặc video. Computer Vision thực hiện các nhiệm vụ như phát hiện khuôn mặt (Face Detection), nhận dạng khuôn mặt (Face Recognition), theo dõi chuyển động, trích xuất đặc trưng... Đây là công nghệ nền tảng quyết định hiệu quả hoạt động của hệ thống điểm danh bằng camera.

### ***1.1.2. Thị giác máy tính (Computer Vision)***

Thị giác máy tính là một lĩnh vực của trí tuệ nhân tạo, tập trung vào việc giúp máy tính nhận biết, phân tích và hiểu nội dung của hình ảnh hoặc video. Công nghệ này cho phép máy tính thực hiện các nhiệm vụ như phát hiện đối tượng, phân loại hình ảnh, theo dõi chuyển động, và đặc biệt là nhận diện khuôn mặt. [2]

Quy trình hoạt động cơ bản của thị giác máy tính gồm các bước:

- Thu thập dữ liệu hình ảnh: thông qua camera hoặc cảm biến.
- Tiền xử lý hình ảnh: bao gồm chuyển đổi kích thước, lọc nhiễu, cân bằng sáng, v.v.
- Phát hiện đối tượng (Object Detection): xác định vị trí của khuôn mặt hoặc vật thể cần nhận diện.
- Trích xuất đặc trưng (Feature Extraction): phân tích các đặc điểm quan trọng như hình dạng, màu sắc, vị trí.
- Phân loại và nhận diện: so sánh với dữ liệu đã học để đưa ra kết quả nhận dạng.

Học máy (Machine Learning) đóng vai trò quan trọng trong sự phát triển của thị giác máy tính (Computer Vision), giúp máy tính có khả năng hiểu, phân tích và trích xuất thông tin từ hình ảnh và video. Khác với các phương pháp xử lý ảnh truyền thống vốn phụ thuộc vào các quy tắc thủ công (rule-based), học máy cho phép mô hình tự học đặc trưng từ dữ liệu, nhờ đó đạt độ chính xác cao hơn và giảm thiểu phụ thuộc vào lập trình viên. [3] Trong đề tài này, thị giác máy tính đóng vai trò trung tâm, giúp hệ thống nhận

diện khuôn mặt sinh viên thông qua hình ảnh được thu từ camera, sau đó xác định danh tính và ghi nhận kết quả điểm danh một cách tự động.

## **1.2. Công nghệ nhận diện khuôn mặt**

### **1.2.1. Lịch sử phát triển**

Công nghệ nhận diện khuôn mặt bắt đầu hình thành từ những năm 1960, khi các nhà khoa học đầu tiên thử nghiệm việc xác định danh tính con người dựa trên đặc điểm khuôn mặt. Năm 1964, Woodrow W. Bledsoe được xem là người tiên phong khi phát triển hệ thống bán tự động sử dụng các điểm đặc trưng trên khuôn mặt để nhận dạng.

Đến thập niên 1990, với sự ra đời của thuật toán Eigenfaces, công nghệ nhận diện khuôn mặt đã có bước tiến vượt bậc khi máy tính có thể tự động phân tích và nhận dạng khuôn mặt dựa trên dữ liệu hình ảnh. Tiếp theo là các phương pháp Fisherfaces và Local Binary Patterns (LBP) giúp cải thiện độ chính xác trong các điều kiện ánh sáng khác nhau.

Từ năm 2010 trở lại đây, sự phát triển mạnh mẽ của trí tuệ nhân tạo (AI) và học sâu (Deep Learning) đã mở ra kỷ nguyên mới cho nhận diện khuôn mặt. Các mô hình như Convolutional Neural Networks (CNN), DeepFace (Facebook), FaceNet (Google) hay VGG-Face đã nâng độ chính xác lên mức tương đương, thậm chí vượt qua khả năng nhận dạng của con người trong một số trường hợp.

Ngày nay, công nghệ nhận diện khuôn mặt được ứng dụng rộng rãi trong nhiều lĩnh vực như bảo mật, ngân hàng, giám sát an ninh, thương mại điện tử và đặc biệt là trong giáo dục, giúp tự động hóa quy trình điểm danh, kiểm soát ra vào và quản lý học viên.

### **1.2.2. Nhận diện khuôn mặt là gì**

Nhận diện khuôn mặt (Face Recognition) là một công nghệ thuộc lĩnh vực trí tuệ nhân tạo (AI) và xử lý ảnh (Image Processing), có khả năng xác định hoặc xác thực danh tính của một người dựa trên các đặc điểm sinh trắc học của khuôn mặt. Hệ thống nhận diện khuôn mặt hoạt động bằng cách phát hiện khuôn mặt trong hình ảnh hoặc video, sau đó trích xuất các đặc trưng như khoảng cách giữa hai mắt, hình dạng mũi, miệng, cằm... và so sánh với dữ liệu khuôn mặt đã được lưu trữ trong cơ sở dữ liệu.

Công nghệ này thường gồm ba giai đoạn chính: phát hiện khuôn mặt (face detection), trích xuất đặc trưng (feature extraction) và so sánh nhận dạng (face matching). Nhờ sự phát triển của học máy (Machine Learning) và học sâu (Deep Learning), đặc biệt là các mạng nơ-ron tích chập (CNN), độ chính xác của hệ thống nhận diện khuôn mặt ngày càng được cải thiện, giúp công nghệ này trở thành một phần quan trọng trong các ứng dụng hiện đại như mở khóa thiết bị, giám sát an ninh và điểm danh tự động.

### **1.2.3. Công nghệ nhận diện khuôn mặt hoạt động như thế nào?**

Hệ thống nhận diện khuôn mặt hoạt động dựa trên quá trình phân tích và so sánh các đặc điểm sinh trắc học của khuôn mặt người. Quá trình này thường trải qua bốn bước chính:

Bước 1 – Phát hiện khuôn mặt (Face Detection): Hệ thống xác định vị trí khuôn mặt trong ảnh hoặc video. Công đoạn này giúp tách phần khuôn mặt ra khỏi nền và các đối tượng khác trong khung hình.

Bước 2 – Căn chỉnh khuôn mặt (Face Alignment): Sau khi phát hiện, khuôn mặt được căn chỉnh về vị trí và góc nhìn chuẩn, giúp việc trích xuất đặc trưng đạt độ chính xác cao hơn.

Bước 3 – Trích xuất đặc trưng (Feature Extraction): Hệ thống phân tích các đặc điểm như khoảng cách giữa hai mắt, hình dáng mũi, cằm, hoặc cấu trúc khuôn mặt. Các đặc trưng này được chuyển đổi thành dữ liệu số (vector đặc trưng) để lưu trữ và so sánh.

Bước 4 – So khớp và nhận dạng (Face Matching): Cuối cùng, hệ thống so sánh vector đặc trưng vừa thu được với dữ liệu khuôn mặt đã có trong cơ sở dữ liệu. Nếu độ tương đồng vượt ngưỡng cho phép, hệ thống xác nhận danh tính của người đó.



*Hình 1: Công nghệ nhận diện khuôn mặt*

Nhờ việc áp dụng các thuật toán học sâu (Deep Learning) và mạng nơ-ron tích chập (CNN), quy trình nhận diện ngày nay đạt tốc độ xử lý nhanh và độ chính xác cao, có thể áp dụng hiệu quả trong nhiều lĩnh vực như bảo mật, quản lý nhân sự, kiểm soát ra vào và điểm danh tự động.

#### **1.2.4. Ứng dụng của công nghệ nhận diện khuôn mặt trong các ngành nghề**

Công nghệ nhận diện khuôn mặt hiện nay được ứng dụng rộng rãi trong nhiều lĩnh vực nhờ khả năng tự động hóa và độ chính xác cao. Trong lĩnh vực an ninh, công nghệ này được sử dụng để giám sát, kiểm soát ra vào và hỗ trợ điều tra tội phạm thông qua hệ thống camera thông minh. Trong ngân hàng và tài chính, nhận diện khuôn mặt giúp xác thực danh tính khách hàng khi thực hiện giao dịch trực tuyến hoặc đăng nhập tài khoản, góp phần tăng cường bảo mật.

Trong thương mại và marketing, công nghệ này hỗ trợ nhận diện hành vi khách hàng, phân tích độ tuổi, giới tính để đưa ra chiến lược quảng cáo phù hợp. Trong chính phủ và hành chính công, nhận diện khuôn mặt được dùng để cấp giấy tờ, quản lý dân cư và kiểm soát biên giới. Đặc biệt trong giáo dục, công nghệ này được ứng dụng trong việc điểm danh sinh viên, kiểm soát ra vào lớp học và đảm bảo tính chính xác, minh bạch trong quản lý. [4]

#### **1.2.5. Ưu và nhược điểm của nhận diện khuôn mặt**

Công nghệ nhận diện khuôn mặt mang lại nhiều lợi ích nổi bật nhưng cũng tồn tại một số hạn chế nhất định trong quá trình ứng dụng thực tế.

Ưu điểm:

- Tính tiện lợi: Người dùng không cần mang theo thẻ hay nhập mật khẩu, chỉ cần khuôn mặt để xác thực danh tính.
- Tốc độ và tự động hóa cao: Quá trình nhận dạng diễn ra nhanh chóng, giúp tiết kiệm thời gian trong các hoạt động như điểm danh, kiểm soát ra vào hay giao dịch trực tuyến.
- Độ chính xác ngày càng cao: Nhờ ứng dụng trí tuệ nhân tạo (AI) và học sâu (Deep Learning), hệ thống có thể nhận diện chính xác ngay cả trong điều kiện thay đổi ánh sáng hoặc góc nhìn.
- Khả năng tích hợp linh hoạt: Có thể kết hợp với các hệ thống khác như camera giám sát, quản lý nhân sự hoặc kiểm soát an ninh.

Nhược điểm:

- Ảnh hưởng bởi điều kiện môi trường: Ánh sáng yếu, góc chụp nghiêng hoặc che khuất khuôn mặt có thể làm giảm độ chính xác.
- Vấn đề bảo mật và quyền riêng tư: Dữ liệu khuôn mặt là thông tin sinh trắc học nhạy cảm, nếu bị lộ có thể gây rủi ro về bảo mật cá nhân.
- Chi phí triển khai cao: Hệ thống yêu cầu phần cứng, camera và máy chủ xử lý mạnh, gây tốn kém cho các đơn vị có quy mô lớn.

- Hạn chế trong việc nhận diện nhóm người đặc biệt: Ví dụ như trẻ nhỏ, người đeo khẩu trang hoặc người có khuôn mặt thay đổi theo thời gian.

Tóm lại, công nghệ nhận diện khuôn mặt là một bước tiến quan trọng trong tự động hóa và quản lý thông minh, tuy nhiên việc triển khai cần được cân nhắc giữa hiệu quả, chi phí và yếu tố bảo mật.

### **1.3. Các thư viện và công cụ hỗ trợ**

#### **1.3.1. Các thư viện chính**

##### **1.3.1.1. OpenCV**

OpenCV (Open Source Computer Vision Library) là một thư viện mã nguồn mở được phát triển nhằm hỗ trợ xử lý ảnh và thị giác máy tính (Computer Vision). Thư viện này được giới thiệu lần đầu tiên vào năm 2000 bởi Intel và hiện được duy trì, phát triển bởi cộng đồng mã nguồn mở trên toàn thế giới. OpenCV cung cấp hàng nghìn hàm xử lý ảnh và video, bao gồm các chức năng như phát hiện khuôn mặt, nhận diện đối tượng, phân tích chuyển động, trích xuất đặc trưng hình ảnh và nhiều thuật toán thị giác máy tính khác. Thư viện này hỗ trợ nhiều ngôn ngữ lập trình như Python, C++, Java và có khả năng hoạt động trên nhiều nền tảng như Windows, Linux, macOS, Android. [5]

Các đặc điểm nổi bật của OpenCV:

- Xử lý ảnh và video mạnh mẽ: OpenCV cung cấp đầy đủ các hàm xử lý ảnh từ cơ bản đến nâng cao như biến đổi màu sắc, lọc nhiễu, phân đoạn ảnh, phát hiện biên, biến đổi hình học, đọc/ghi ảnh và luồng video, đáp ứng tốt cho các ứng dụng thời gian thực.
- Hệ thống mô-đun phong phú.
- Hỗ trợ Deep Learning: Mô-đun DNN cho phép triển khai các mạng neural dùng cho phát hiện và nhận diện đối tượng hoặc mô hình nhận diện khuôn mặt hiện đại.
- Tương thích đa nền tảng và đa ngôn ngữ lập trình: Hỗ trợ Python, C++, Java, và chạy được trên nhiều hệ điều hành như Windows, Linux, macOS, Android, iOS, giúp dễ dàng tích hợp vào các hệ thống phần mềm.
- Cộng đồng lớn và tài liệu phong phú: Với hàng triệu người dùng trên toàn thế giới, OpenCV có kho tài liệu, ví dụ, diễn đàn hỗ trợ phong phú, giúp dễ dàng học và triển khai.
- Khả năng kết hợp với các thư viện khác: Tương thích tốt với NumPy, TensorFlow, PyTorch,... giúp mở rộng khả năng xây dựng các hệ thống AI phức tạp.

- Ứng dụng rộng rãi trong thực tế: OpenCV được sử dụng trong nhiều lĩnh vực như giám sát an ninh, xe tự hành, y tế, robot và đặc biệt là nhận diện khuôn mặt trong hệ thống điểm danh tự động.

Trong đề tài website điểm danh bằng nhận diện khuôn mặt, OpenCV được sử dụng để phát hiện và xử lý khuôn mặt từ camera hoặc ảnh, đóng vai trò quan trọng trong việc thu thập và chuẩn bị dữ liệu cho bước nhận diện. Nhờ tính linh hoạt, hiệu năng cao và cộng đồng phát triển lớn, OpenCV trở thành công cụ phổ biến và đáng tin cậy trong lĩnh vực xử lý ảnh và trí tuệ nhân tạo.

#### *1.3.1.2. PyTorch*

PyTorch là một framework học sâu (Deep Learning) mã nguồn mở được phát triển bởi Facebook AI Research (FAIR). Ra mắt vào năm 2016, PyTorch nhanh chóng trở thành một trong những nền tảng phổ biến nhất để xây dựng và huấn luyện các mô hình học máy nhờ tính linh hoạt, cú pháp thân thiện và khả năng xử lý mạnh mẽ trên GPU.

PyTorch hỗ trợ hai thành phần quan trọng:

- Tensor: Là cấu trúc dữ liệu đa chiều tương tự NumPy nhưng có khả năng chạy trên GPU. Tensor trong PyTorch cho phép xử lý song song và tối ưu hiệu năng trong các phép toán ma trận – yếu tố quan trọng trong Deep Learning.
- Autograd: Là hệ thống tự động tính đạo hàm, gradient cho các mô hình mạng nơ-ron. Autograd giúp quá trình huấn luyện mô hình trở nên đơn giản và hiệu quả, giảm đáng kể thời gian triển khai các thuật toán tối ưu hóa.

Các đặc điểm và ưu điểm nổi bật của PyTorch:

- Dễ sử dụng và thân thiện với lập trình viên: PyTorch cho phép xây dựng mô hình theo hướng “dynamic graph” (đồ thị động), dễ sửa lỗi, dễ debug và phù hợp cho cả nghiên cứu lẫn phát triển sản phẩm.
- Hỗ trợ GPU mạnh mẽ: Các phép toán Tensor được tăng tốc đáng kể nhờ hỗ trợ CUDA, giúp mô hình Deep Learning đạt hiệu năng xử lý cao trong các ứng dụng real-time.
- Linh hoạt trong tùy chỉnh mô hình: Cho phép dễ dàng xây dựng, chỉnh sửa và mở rộng các mô hình mạng nơ-ron phức tạp.
- Kho thư viện phong phú: Bao gồm TorchVision, TorchAudio, TorchText,... phục vụ nhiều bài toán AI khác nhau.
- Cộng đồng lớn và tài liệu đầy đủ: PyTorch được sử dụng rộng rãi trong nghiên cứu khoa học và công nghiệp, giúp người dùng dễ dàng học tập và khắc phục vấn đề.

- Tương thích tốt với NumPy và các thư viện Python khác: Giúp kết hợp trong pipeline xử lý ảnh – trích xuất đặc trưng – nhận diện khuôn mặt một cách thuận tiện.

Vai trò của PyTorch trong hệ thống nhận diện khuôn mặt: Triển khai mô hình ArcFace + iResNet-100 giúp trích xuất đặc trưng khuôn mặt dưới dạng vector 512 chiều, tăng tốc độ xử lý bằng GPU giúp mô hình chạy nhanh hơn gấp nhiều lần so với CPU, xử lý batch ảnh và vector embedding hỗ trợ pipeline nhận diện khi nhiều khuôn mặt xuất hiện cùng lúc, tích hợp với TorchVision để tiền xử lý ảnh.

Nhìn chung, PyTorch là nền tảng học sâu mạnh mẽ, linh hoạt và phù hợp cho cả nghiên cứu lẫn triển khai thực tế. Với khả năng hỗ trợ GPU hiệu quả, hệ thống autograd tiên tiến và cấu trúc Tensor tối ưu, PyTorch đáp ứng tốt các yêu cầu của những mô hình xử lý ảnh và nhận diện khuôn mặt hiện đại. Trong phạm vi đề tài, PyTorch đóng vai trò quan trọng trong việc xây dựng mô hình trích xuất đặc trưng khuôn mặt có độ chính xác cao, đồng thời đảm bảo tốc độ xử lý nhanh, ổn định khi tích hợp với hệ thống điểm danh. Việc sử dụng PyTorch không chỉ nâng cao hiệu quả nhận diện của mô hình mà còn giúp hệ thống đạt được tính mở rộng, dễ bảo trì và dễ nâng cấp trong tương lai.

#### *1.3.1.3. NumPy*

NumPy (Numerical Python) là một thư viện xử lý số học mạnh mẽ và phổ biến trong Python, được thiết kế để hỗ trợ các phép toán trên mảng nhiều chiều (ndarray) với hiệu suất cao. Thư viện này được tối ưu hóa bằng cách sử dụng các thuật toán xử lý vector hóa và được viết một phần bằng C, giúp tốc độ xử lý nhanh hơn rất nhiều lần so với các vòng lặp Python thông thường. NumPy đóng vai trò nền tảng cho nhiều thư viện khoa học và học máy khác như OpenCV, PyTorch, Pandas... nhờ khả năng thao tác dữ liệu linh hoạt và ổn định.

Vai trò của NumPy trong hệ thống:

- Xử lý dữ liệu ảnh dưới dạng mảng số: Mỗi bức ảnh thu từ camera được biểu diễn bằng ma trận giá trị điểm ảnh (pixel). NumPy giúp đọc, chuyển đổi và thao tác dữ liệu này một cách hiệu quả.
- Phối hợp với OpenCV và PyTorch: OpenCV trả ảnh dạng mảng NumPy, sau đó có thể chuyển đổi trực tiếp thành Tensor để đưa vào mô hình PyTorch. Điều này giúp quá trình tiền xử lý – suy luận – trích xuất đặc trưng diễn ra liền mạch.
- Tính toán vector đặc trưng (feature embedding): Sau khi PyTorch tạo ra vector embedding 512 chiều của khuôn mặt, NumPy được dùng để chuẩn hóa vector – tính khoảng cách cosine – so sánh mức tương đồng giữa hai khuôn mặt

- Xử lý danh sách dữ liệu và ma trận: Từ dữ liệu khuôn mặt, mô hình cần thực hiện nhiều phép toán ma trận như cộng, trừ, nhân, chuẩn hóa, tìm cực trị,... NumPy giúp tăng tốc và đảm bảo độ chính xác.

Ưu điểm nổi bật của NumPy:

- Hiệu năng cao: Tích hợp các thuật toán tối ưu hóa và xử lý vector hóa giúp tốc độ nhanh hơn nhiều lần so với Python thuần.
- Kết hợp tốt với các thư viện AI: NumPy là nền tảng quan trọng của các thư viện như PyTorch, TensorFlow, OpenCV, hỗ trợ pipeline AI mượt mà.
- API rõ ràng, dễ dùng: Cú pháp đơn giản, trực quan, giúp lập trình viên dễ dàng thực hiện các thao tác phức tạp trên mảng và ma trận.
- Ổn định và đáng tin cậy: Là thư viện chuẩn của cộng đồng khoa học dữ liệu và AI, được sử dụng rộng rãi trong nghiên cứu, học thuật và công nghiệp.

NumPy đóng vai trò như “xương sống” trong xử lý dữ liệu của hệ thống nhận diện khuôn mặt. Nhờ cung cấp các công cụ tính toán hiệu năng cao, NumPy giúp tối ưu hóa tốc độ xử lý ảnh, hỗ trợ các thư viện như OpenCV và PyTorch hoạt động hiệu quả, đồng thời đảm bảo độ chính xác trong các phép tính toán vector đặc trưng. Sự ổn định, tốc độ và khả năng tích hợp mạnh mẽ của NumPy khiến thư viện này trở thành thành phần quan trọng và không thể thiếu trong toàn bộ pipeline điểm danh bằng nhận diện khuôn mặt.

#### 1.3.1.4. TorchVision

TorchVision là một thư viện mở rộng của PyTorch, được thiết kế chuyên cho xử lý ảnh và xây dựng các mô hình thị giác máy tính. Thư viện cung cấp nhiều công cụ quan trọng như tập dữ liệu mẫu (datasets), các mô hình pretrained, và đặc biệt là bộ transform dùng để tiền xử lý ảnh. TorchVision hỗ trợ thao tác ảnh nhanh và chuẩn hóa dữ liệu theo đúng yêu cầu của các mô hình Deep Learning, đảm bảo ảnh đầu vào luôn có định dạng thống nhất và phù hợp. TorchVision hoạt động chặt chẽ với PyTorch, cho phép chuyển đổi ảnh từ dạng NumPy hoặc PIL sang tensor, thay đổi kích thước, chuẩn hóa giá trị pixel và áp dụng các phép biến đổi cần thiết trong pipeline nhận diện khuôn mặt.

Vai trò của TorchVision trong hệ thống:

- Tiền xử lý ảnh đầu vào: Chuyển đổi và chuẩn hóa ảnh trước khi đưa vào mô hình ArcFace hoặc iResNet-100.
- Chuyển đổi từ ảnh sang Tensor: Cho phép kết nối trực tiếp giữa dữ liệu ảnh (OpenCV/NumPy) với mô hình PyTorch.
- Đảm bảo tính thống nhất của dữ liệu: Áp dụng các bước transform như Resize(), CenterCrop(), Normalize() để ảnh đầu vào đạt cùng kích thước và phân phối màu.

- Hỗ trợ thao tác nhanh trong pipeline real-time: Các phép biến đổi ảnh được tối ưu giúp tốc độ xử lý nhanh, phù hợp với yêu cầu điểm danh thời gian thực.
- Tích hợp mượt mà với PyTorch: Cho phép mô hình nhận diện tận dụng trực tiếp các tiện ích mà không cần viết lại thủ công.

Ưu điểm của TorchVision:

- Tích hợp trực tiếp với PyTorch, hỗ trợ liền mạch giữa xử lý ảnh và mô hình học sâu.
- Cung cấp bộ transform mạnh mẽ, tối ưu cho tiền xử lý ảnh trong Deep Learning.
- Giúp chuẩn hóa ảnh đầu vào, đảm bảo mô hình nhận diện hoạt động ổn định và chính xác.
- Tốc độ nhanh, phù hợp cho xử lý real-time trong môi trường camera.
- Hỗ trợ nhiều mô hình thị giác máy tính có sẵn, giúp rút ngắn thời gian phát triển.

TorchVision là thành phần không thể thiếu trong hệ thống nhận diện khuôn mặt sử dụng PyTorch. Nhờ khả năng tiền xử lý ảnh mạnh mẽ, hỗ trợ đa dạng mô hình có sẵn và khả năng tích hợp sâu với PyTorch, TorchVision giúp xây dựng pipeline nhận diện khuôn mặt trở nên đơn giản, mượt mà và hiệu quả hơn. Việc sử dụng thư viện này góp phần nâng cao độ chính xác của mô hình và tăng tốc độ xử lý, từ đó cải thiện trải nghiệm tổng thể của hệ thống điểm danh tự động.

### **1.3.2. Các công cụ hỗ trợ nhận dạng khuôn mặt**

#### **1.3.2.1. SCRFD (Sparsely Connected Regression for Face Detection)**

SCRFD là một mô hình phát hiện khuôn mặt hiện đại thuộc bộ công cụ InsightFace, được thiết kế để đạt tốc độ cao và độ chính xác tốt trong nhiều điều kiện môi trường khác nhau. Điểm mạnh của SCRFD là khả năng phát hiện khuôn mặt nhanh ngay cả trên các thiết bị có cấu hình không quá mạnh. Mô hình được tối ưu cho chạy real-time với camera, phù hợp cho các hệ thống cần xử lý khuôn mặt liên tục như điểm danh, giám sát hoặc kiểm soát ra vào. [6]

SCRFD hoạt động dựa trên cơ chế regression nhẹ, sử dụng mạng nơ-ron được tối ưu hóa để xác định vùng khuôn mặt trong ảnh và trả về tọa độ bounding box chính xác. Nhờ đó, mô hình có thể phát hiện khuôn mặt chính xác ngay cả khi ảnh có góc nghiêng nhẹ, thay đổi ánh sáng hoặc có nhiều đối tượng trong khung hình.

Vai trò của SCRFD trong hệ thống:

- Phát hiện khuôn mặt trong khung hình: SCRFD nhận ảnh đầu vào từ camera và xác định vị trí khuôn mặt bằng bounding box. Đây là bước đầu tiên của pipeline nhận diện, giúp tách vùng chứa khuôn mặt để xử lý tiếp.

- Chuẩn bị dữ liệu cho mô hình ArcFace: SCRFD kết hợp với bước căn chỉnh (alignment) để tạo ảnh khuôn mặt chuẩn, đảm bảo phù hợp với yêu cầu của mô hình ArcFace/iResNet-100. Ảnh sau khi được cắt từ SCRFD sẽ được resize, chuẩn hóa và đưa sang bước trích xuất đặc trưng.
- Xử lý tốt trong môi trường thực tế: SCRFD hoạt động ổn định với nhiều điều kiện ánh sáng, góc mặt và khoảng cách. Điều này giúp hệ thống điểm danh giữ được độ chính xác cao ngay cả khi sinh viên đứng xa camera hoặc di chuyển nhẹ.
- Tốc độ cao, phù hợp chạy real-time: Mô hình chạy nhanh, tiêu tốn ít tài nguyên, giúp hệ thống có thể xử lý nhiều khung hình mỗi giây mà không bị giật lag. Tốc độ này rất quan trọng đối với hệ thống điểm danh tự động.

Ưu điểm của SCRFD:

- Phát hiện khuôn mặt nhanh, phù hợp real-time.
- Hoạt động tốt trên nhiều loại thiết bị Độ chính xác cao ngay cả trong môi trường ánh sáng không ổn định.
- Dễ tích hợp với OpenCV và InsightFace.
- Trả bounding box ổn định giúp cải thiện chất lượng ảnh đầu vào cho mô hình nhận dạng.

Tóm lại, SCRFD là một trong những mô hình phát hiện khuôn mặt hiện đại mang lại hiệu năng vượt trội về tốc độ và độ chính xác. Nhờ cấu trúc mạng được tối ưu hóa và thiết kế gọn nhẹ, SCRFD có khả năng nhận diện khuôn mặt trong thời gian thực ngay cả trên các thiết bị có cấu hình trung bình. Mô hình này không chỉ phát hiện chính xác khuôn mặt trong nhiều điều kiện ánh sáng và góc nhìn khác nhau, mà còn hoạt động ổn định khi có nhiều khuôn mặt xuất hiện đồng thời trong khung hình.

#### *1.3.2.2. ArcFace kết hợp mô hình iResNet-100*

ArcFace là một thuật toán nhận diện khuôn mặt dựa trên học sâu, sử dụng hàm mất mát Additive Angular Margin Loss để tạo ra các vector đặc trưng (embedding) có khả năng phân biệt rất tốt giữa các khuôn mặt khác nhau. Thuật toán này giúp tạo ra khoảng cách góc lớn hơn giữa các lớp, từ đó tăng độ chính xác khi so khớp khuôn mặt. ArcFace hiện là một trong những phương pháp nhận diện khuôn mặt có độ chính xác cao nhất và được sử dụng rộng rãi trong các hệ thống thực tế.

Mô hình backbone được sử dụng trong đề tài là iResNet-100, một phiên bản cải tiến của ResNet với cấu trúc sâu 100 lớp, hỗ trợ trích xuất đặc trưng mạnh và ổn định. iResNet-100 có khả năng học được các đặc trưng phức tạp của khuôn mặt như hình dạng, vị trí, góc nhìn, từ đó tạo ra vector đặc trưng có độ tin cậy cao.

Vai trò của ArcFace + iResNet-100 trong hệ thống:

- Xác định danh tính sinh viên: Hệ thống dùng khoảng cách cosine để so sánh embedding của khuôn mặt hiện tại với embedding đã lưu. Nếu độ tương đồng vượt ngưỡng, hệ thống xác nhận danh tính sinh viên.
- Hoạt động ổn định với ảnh đa dạng: ánh sáng thay đổi, góc mặt lệch nhẹ, biểu cảm khác nhau, chất lượng camera không cao.
- Tích hợp tốt với pipeline OpenCV → TorchVision → PyTorch: Ảnh được phát hiện bởi SCRFD, tiền xử lý bằng TorchVision và đưa vào mô hình ArcFace chạy trên PyTorch để tạo embedding. Pipeline này đảm bảo quá trình nhận diện nhanh và chính xác.

Ưu điểm của ArcFace + iResNet-100:

- Độ chính xác cao, được chứng minh trong nhiều benchmark.
- Vector đặc trưng ổn định và dễ so sánh.
- Hoạt động tốt trong môi trường thực tế.
- Tốc độ xử lý nhanh khi chạy trên GPU.
- Phù hợp thực thi real-time trong ứng dụng điểm danh.

ArcFace kết hợp với backbone iResNet-100 là giải pháp tối ưu cho bài toán nhận diện khuôn mặt trong hệ thống điểm danh. Nhờ cơ chế Additive Angular Margin Loss và khả năng trích xuất đặc trưng mạnh mẽ của mạng iResNet-100, mô hình tạo ra các vector embedding có độ phân tách cao, giúp tăng độ chính xác khi xác định danh tính sinh viên. Bộ đôi này hoạt động ổn định trong nhiều điều kiện thực tế như thay đổi ánh sáng, góc nhìn hay chất lượng camera khác nhau, đồng thời đạt hiệu năng tốt khi triển khai real-time. Việc tích hợp vào pipeline cùng SCRFD, OpenCV và PyTorch giúp hệ thống đạt độ tin cậy, tốc độ và tính nhất quán cao, đáp ứng tốt yêu cầu của bài toán điểm danh bằng nhận diện khuôn mặt.

### *1.3.2.3. Git*

Git là một hệ thống quản lý phiên bản phân tán (Distributed Version Control System – DVCS) rất được ưa chuộng, được sử dụng để theo dõi và quản lý các sự thay đổi trong mã nguồn trong suốt quá trình phát triển phần mềm. Được tạo ra bởi Linus Torvalds vào năm 2005, Git đã nhanh chóng trở thành một trong những công cụ quản lý phiên bản phổ biến và quan trọng nhất trên toàn cầu. Git dễ học và có dung lượng nhỏ với hiệu suất cực nhanh. Nó vượt trội hơn các công cụ SCM như Subversion, CVS, Perforce và ClearCase với các tính năng như tạo nhánh cục bộ giá rẻ, khu vực tạm thời tiện lợi và nhiều quy trình làm việc. Mục đích chính của Git là giúp đội ngũ phát triển làm việc cùng nhau trên cùng một dự án mà không gặp khó khăn về việc ghi đè lên công việc của nhau. Nó cho phép mọi người làm việc trên các phiên bản riêng của mã nguồn và sau đó hợp nhất các thay đổi này thành phiên bản chính thức, được gọi là "nhánh chính" (thường được gọi là "master" hoặc "main").



*Hình 2: Khái niệm công cụ Git*

Các thao tác sử dụng Git:

- Tạo và quản lý kho lưu trữ dự án.
- Tạo branch riêng cho từng chức năng như đăng ký, đăng nhập, quản lý tour, đặt tour...
- Thực hiện commit thường xuyên kèm thông điệp rõ ràng để theo dõi tiến độ.
- Đẩy (push) mã nguồn lên GitHub để lưu trữ và chia sẻ với mentor, đồng đội.
- Sử dụng pull request để hợp nhất code và nhận góp ý từ người hướng dẫn.
- Giải quyết xung đột (conflict) khi có thay đổi trùng nhau giữa các nhánh.

Lợi ích của Git đem lại:

- Version Control: Git giúp theo dõi thay đổi mã nguồn theo thời gian, cho phép quay trở lại phiên bản trước và hiểu lịch sử sửa đổi.
- Collaboration: Cho phép nhiều nhà phát triển cùng làm việc trên dự án, tạo nhánh riêng để xử lý tính năng hoặc lỗi, sau đó hợp nhất vào mã nguồn chính để phát triển đồng thời.

- Branching and Experimentation: Hệ thống nhánh (branch) của Git cho phép tạo các branch độc lập cho tính năng mới, bảo vệ mã nguồn chính. Nếu thành công, nhánh này có thể hợp nhất (merge) vào nhánh chính.
- Conflict Resolution: Git giúp giải quyết xung đột khi nhiều nhà phát triển cùng sửa mã nguồn. Công cụ của Git đảm bảo tích hợp thay đổi không xảy ra xung đột.
- Traceability: Git lưu lịch sử chi tiết thay đổi code, bao gồm tác giả, thời gian và lý do. Khả năng này hữu ích để hiểu lịch sử thay đổi code và khắc phục lỗi.
- Code Reviews: Nền tảng Git như GitHub và GitLab,... hỗ trợ code review. Lập trình viên có thể tạo các pull request, sau đó người khác review code, cải thiện chất lượng code và chia sẻ kiến thức.
- Rollbacks and Bug Fixes: Khi gặp lỗi hoặc thay đổi không mong muốn, Git cho phép quay trở lại phiên bản trước ổn định, giảm nguy cơ lỗi và tiết kiệm thời gian.
- Backup and Redundancy: Mỗi nhà phát triển có bản sao đầy đủ của mã nguồn, giảm rủi ro mất dữ liệu. Các bản sao được lưu ở git server.
- Offline Work: Git có tính phân tán, cho phép nhà phát triển làm việc ngoại tuyến, hỗ trợ nhóm làm việc từ xa hoặc phân tán.
- Open Source and Collaboration: Git là mã nguồn mở và được sử dụng rộng rãi. Nhiều dự án mã nguồn mở được lưu trữ trên các nền tảng như GitHub, giúp các nhà phát triển đóng góp và hợp tác trên các dự án khác nhau một cách dễ dàng.
- Scalability: Git có thể xử lý các dự án với các kích thước khác nhau, từ các dự án cá nhân nhỏ đến các dự án mã nguồn cấp doanh nghiệp lớn.
- Custom Workflows: Có thể tùy chỉnh workflow linh hoạt dựa trên git, tạo ra các branch, các release khác nhau theo nhu cầu cụ thể.

#### *1.3.2.4. Github*

GitHub là nền tảng lưu trữ và quản lý mã nguồn sử dụng hệ thống quản lý phiên bản Git. Trong dự án, GitHub được sử dụng như một kho lưu trữ (repository) trung tâm, giúp nhóm phát triển quản lý mã nguồn, theo dõi tiến độ và phối hợp làm việc hiệu quả. GitHub cho phép lưu lại toàn bộ lịch sử thay đổi của mã nguồn, hỗ trợ kiểm soát phiên bản, khôi phục khi gặp lỗi và đảm bảo tính đồng bộ giữa các thành viên. Thông qua các tính năng như Branch, Commit, Pull Request và Issues, nhóm có thể dễ dàng phân chia công việc, phát triển song song nhiều tính năng và kiểm tra, rà soát mã trước khi tích hợp vào hệ thống chính.

Vai trò của GitHub:

- Quản lý phiên bản mã nguồn: Theo dõi lịch sử thay đổi, giúp dễ dàng xem lại, khôi phục hoặc so sánh các phiên bản khi cần.

- Đảm bảo tính ổn định của dự án: Mỗi tính năng hoặc chỉnh sửa đều được đưa vào nhánh riêng (branch), giúp tránh ảnh hưởng đến phiên bản ổn định.
- Tạo môi trường backup an toàn: Mã nguồn được lưu trữ trên cloud, tránh mất dữ liệu khi máy tính cá nhân gặp sự cố.
- Tích hợp với các công cụ tự động hóa: Hỗ trợ CI/CD, dễ dàng triển khai bản cập nhật hệ thống nếu cần.

Lợi ích của GitHub:

- Phân chia công việc rõ ràng: Mỗi thành viên có thể làm việc trên nhánh riêng (feature branch), giảm xung đột mã nguồn và giúp quản lý công việc minh bạch.
- Dễ dàng kiểm tra và rà soát mã nguồn: Tính năng Pull Request cho phép các thành viên xem xét, góp ý hoặc phê duyệt mã trước khi hợp nhất (merge), đảm bảo chất lượng mã tốt hơn.
- Theo dõi tiến độ và nhiệm vụ: GitHub Issues và Projects giúp tạo, gán, theo dõi công việc, đặc biệt hiệu quả đối với nhóm làm đồ án.
- Đồng bộ và tránh lỗi trùng sửa mã: GitHub đảm bảo mọi thành viên luôn sử dụng phiên bản mã nguồn mới nhất thông qua git pull, tránh chồng chéo hoặc mất code.
- Giao tiếp và trao đổi ngay trong dự án: Các bình luận, thảo luận ngay trên Pull Request giúp nhóm trao đổi dễ dàng, không cần dùng nhiều công cụ chat khác.
- Lưu trữ an toàn và truy cập mọi lúc: Toàn bộ dự án được sao lưu trên GitHub, có thể truy cập từ bất cứ đâu, trên bất kỳ máy tính nào.

#### *1.3.2.5. XAMPP*

XAMPP là một gói phần mềm tích hợp gồm Apache, MySQL (MariaDB), PHP và Perl, được sử dụng rộng rãi để xây dựng và chạy môi trường máy chủ (server) ngay trên máy tính cá nhân. Nhờ khả năng cài đặt đơn giản, cấu hình dễ dàng và hoạt động ổn định, XAMPP thường được sử dụng để phát triển và kiểm thử các ứng dụng web trước khi triển khai thực tế.

Đặc điểm của XAMPP:

- Dễ cài đặt và sử dụng: XAMPP cung cấp trình cài đặt nhanh, cấu hình đơn giản, phù hợp cho sinh viên và nhà phát triển.
- Tích hợp nhiều thành phần: Bao gồm Apache, MySQL/MariaDB, PHP, Perl – giúp thiết lập đầy đủ môi trường phát triển web chỉ với một bộ cài.
- Hoạt động đa nền tảng: Hỗ trợ Windows, Linux, macOS.
- Giao diện quản lý trực quan: Bật/tắt dịch vụ dễ dàng qua XAMPP Control Panel.
- Thích hợp cho phát triển cục bộ (local): Cho phép chạy và kiểm thử ứng dụng web ngay trên máy tính mà không cần server thật.

XAMPP là công cụ hiệu quả và cần thiết trong quá trình phát triển hệ thống điểm danh. Nhờ tích hợp sẵn Apache và MySQL, XAMPP giúp nhóm phát triển thiết lập môi trường backend và cơ sở dữ liệu một cách nhanh chóng, thuận tiện và ổn định. Môi trường local đồng nhất giúp hạn chế lỗi phát sinh, hỗ trợ kiểm thử, đảm bảo tính chính xác trước khi triển khai chính thức lên server thực tế.

#### **1.4. Công nghệ phát triển web**

##### **1.4.1. Python**

Python là ngôn ngữ lập trình chính được sử dụng để xây dựng phần backend của hệ thống, đặc biệt trong các tác vụ xử lý hình ảnh, chạy mô hình nhận diện khuôn mặt và giao tiếp giữa website và mô hình AI. Python được lựa chọn nhờ cú pháp đơn giản, dễ đọc, dễ vận hành và sở hữu hệ sinh thái thư viện rất mạnh trong lĩnh vực trí tuệ nhân tạo, học máy và xử lý ảnh.

Vai trò Python trong hệ thống:

- Xử lý hình ảnh từ camera: Python kết hợp với thư viện OpenCV để tiếp nhận khung hình từ camera, thực hiện tiền xử lý như chuyển đổi màu, căn chỉnh ảnh và chuẩn bị dữ liệu cho mô hình nhận diện.
- Tích hợp và chạy mô hình nhận diện khuôn mặt: Các mô hình như SCRFD và ArcFace + iResNet-100 được triển khai bằng Python nhờ khả năng kết hợp tốt với PyTorch, TorchVision và NumPy.
- Quản lý dữ liệu và xử lý logic nghiệp vụ: Python sử dụng các framework như Flask hoặc FastAPI để xây dựng các API giúp website gửi ảnh lên server, nhận kết quả nhận diện và lưu trữ bản ghi điểm danh.
- Xây dựng API giao tiếp với frontend: Python đảm nhiệm việc xử lý điểm danh, lưu dữ liệu vào cơ sở dữ liệu, quản lý thông tin lớp học, sinh viên và buổi học.

Các đặc điểm nổi bật của python:

- Cú pháp đơn giản: Dễ đọc, dễ học, gần gũi với ngôn ngữ tự nhiên (tiếng Anh), giúp tập trung vào giải quyết vấn đề thay vì vật lộn với cú pháp.
- Mã nguồn mở và miễn phí: Cộng đồng hỗ trợ cực lớn.
- Đa nền tảng (Portable): Chạy được trên Windows, Linux, macOS mà không cần thay đổi code nhiều.
- Hỗ trợ đa dạng: Lập trình hướng đối tượng (OOP), lập trình hàm, lập trình thủ tục.

Nhờ khả năng xử lý ảnh mạnh mẽ, tích hợp tốt với các mô hình AI, hỗ trợ thư viện phong phú và khả năng triển khai API nhanh chóng, Python là lựa chọn tối ưu để xây dựng backend cho hệ thống điểm danh bằng nhận diện khuôn mặt trong đề tài. Ngôn

ngữ này giúp đảm bảo hệ thống hoạt động ổn định, dễ mở rộng, dễ bảo trì và đạt hiệu suất cao trong môi trường thực tế.

#### **1.4.2. Flask**

Flask là một micro-framework Python dùng để xây dựng ứng dụng web và API. Với thiết kế nhẹ, linh hoạt và dễ mở rộng, Flask đặc biệt phù hợp cho các hệ thống cần tốc độ xử lý nhanh và khả năng tích hợp với các mô hình trí tuệ nhân tạo. Trong đề tài này, Flask được sử dụng để triển khai backend, quản lý giao tiếp giữa giao diện website và mô hình nhận diện khuôn mặt. Flask hoạt động theo mô hình request-response, cho phép frontend gửi dữ liệu (ví dụ: ảnh từ camera) lên server, sau đó server xử lý và trả về kết quả nhận diện. Nhờ đó, Flask đóng vai trò trung tâm trong pipeline điểm danh. [7]



*Hình 3: Khái niệm về Flask*

Vai trò của Flask trong hệ thống:

- Xây dựng API giao tiếp với frontend: Frontend gửi ảnh lên server qua phương thức POST; Flask xử lý, chạy mô hình và trả kết quả JSON.
- Tích hợp mô hình AI (SCRFD, ArcFace + iResNet-100): Flask load mô hình vào RAM/GPU một lần, giúp tăng tốc độ phản hồi mỗi lần nhận diện.
- Xử lý nghiệp vụ điểm danh: Ghi nhận thời gian, lớp học, mã sinh viên và lưu vào cơ sở dữ liệu.
- Kết nối cơ sở dữ liệu: Flask làm việc với CSDL thông qua các ORM hoặc thư viện kết nối.

Ưu điểm của Flask:

- Nhẹ, dễ triển khai và phù hợp hệ thống AI thời gian thực.
- Linh hoạt, dễ mở rộng bằng cách cài thêm extension.
- Hiệu năng cao, có khả năng xử lý nhiều request/giây.
- Dễ tích hợp với OpenCV, PyTorch, NumPy và các thư viện xử lý ảnh.

- Cấu trúc đơn giản, dễ quản lý và bảo trì.

Các đặc điểm nổi bật của Flask:

- Micro-framework đúng nghĩa: chỉ cung cấp những thành phần tối thiểu, cho phép lập trình viên tự xây dựng phần còn lại.
- Routing đơn giản nhưng mạnh mẽ: dễ dàng tạo các API nhận diện khuôn mặt.
- Hỗ trợ middleware và decorator: giúp logging, xác thực, chống spam request, bảo mật API.
- Khả năng xử lý bất đồng bộ (khi dùng với gevent, gunicorn, uvicorn): tăng tốc độ khi có nhiều người dùng điểm danh cùng lúc.
- Dễ triển khai thành microservice: phù hợp khi muốn tách riêng service nhận diện thành API độc lập.

Nhờ sự đơn giản nhưng mạnh mẽ, Flask đáp ứng tốt yêu cầu của một hệ thống điểm danh sử dụng camera và AI, đảm bảo tốc độ xử lý nhanh và tính tương tác thời gian thực giữa người dùng và máy chủ.

#### **1.4.3. MySQL**

MySQL là hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System – RDBMS) mã nguồn mở phổ biến nhất hiện nay, được sử dụng rộng rãi trong các ứng dụng web nhờ khả năng lưu trữ dữ liệu ổn định, tốc độ xử lý nhanh và hỗ trợ tốt cho các hệ thống lớn. MySQL sử dụng ngôn ngữ truy vấn SQL (Structured Query Language) để thao tác dữ liệu như thêm, sửa, xóa và truy vấn, giúp quá trình quản lý dữ liệu trở nên trực quan và dễ dàng.

Đặc điểm nổi bật của MySQL:

- Mã nguồn mở: miễn phí, dễ tùy chỉnh và triển khai.
- Hiệu năng cao: tối ưu truy vấn nhanh, phù hợp với hệ thống có lượng dữ liệu lớn.
- Dễ tích hợp: tương thích tốt với Python, Flask, Node.js và các framework web phổ biến.
- Bảo mật tốt: hỗ trợ phân quyền người dùng, mã hóa dữ liệu và sao lưu tự động.
- Hỗ trợ nhiều loại dữ liệu: INT, VARCHAR, DATE, BLOB, JSON...

Vai trò của MySQL trong hệ thống:

- Quản lý sinh viên: mã sinh viên, họ tên, lớp, ảnh để nhận diện.
- Quản lý lớp học: mã lớp, tên lớp,...
- Dữ liệu điểm danh: lịch sử điểm danh, thời gian điểm danh, trạng thái sinh viên.
- Tài khoản người dùng: giảng viên, quản trị viên.
- Hỗ trợ báo cáo thống kê.

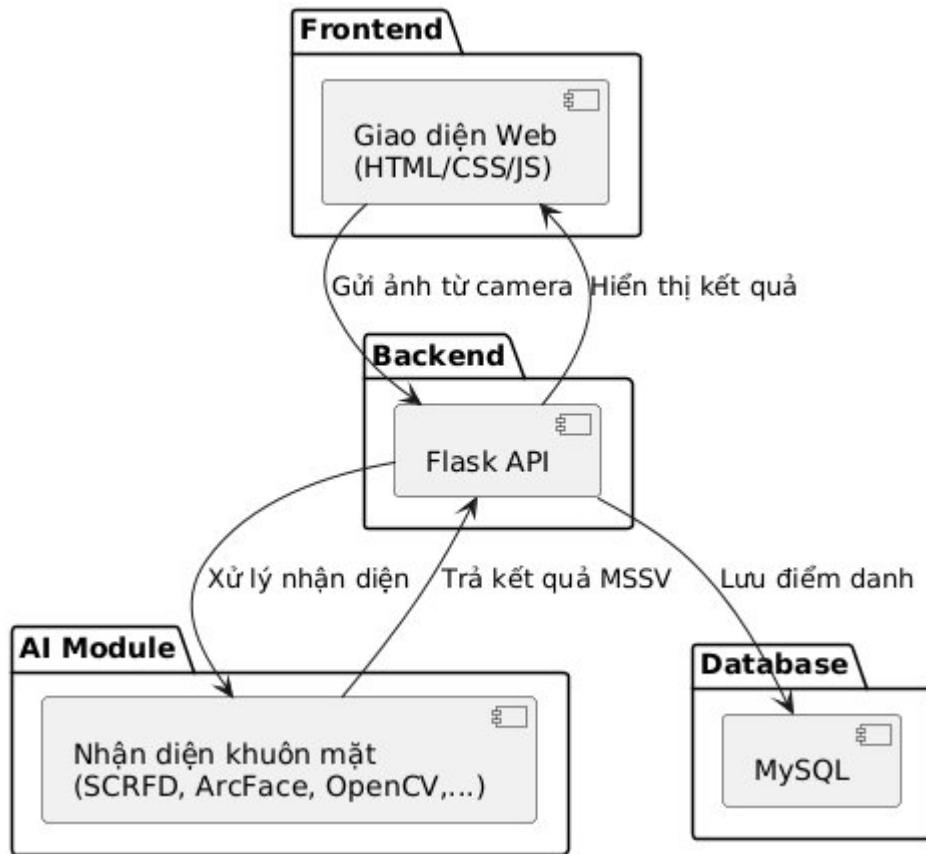
MySQL là lựa chọn phù hợp cho hệ thống điểm danh bằng nhận diện khuôn mặt nhờ khả năng lưu trữ ổn định, hỗ trợ truy vấn mạnh mẽ và dễ tích hợp với Flask và Python. Cơ sở dữ liệu đóng vai trò quan trọng trong việc đảm bảo tính chính xác, an toàn và khả năng mở rộng của hệ thống.

### **1.5. Kiến trúc hệ thống web ứng dụng trí tuệ nhân tạo**

Hệ thống điểm danh sinh viên bằng nhận diện khuôn mặt được xây dựng theo mô hình kiến trúc gồm ba lớp chính: Frontend, Backend và Hệ thống trí tuệ nhân tạo (AI Processing). Mỗi lớp đảm nhận các chức năng độc lập nhưng liên kết chặt chẽ nhằm đảm bảo toàn bộ quy trình điểm danh diễn ra tự động, chính xác và thời gian thực. Ngoài ra, hệ thống còn tích hợp cơ sở dữ liệu MySQL để lưu trữ và truy xuất thông tin một cách ổn định và hiệu quả.

Kiến trúc của hệ thống bao gồm các thành phần chính:

- Frontend (Giao diện người dùng): Xây dựng bằng HTML, CSS, JavaScript. Hiển thị giao diện điểm danh, giao diện quản lý sinh viên – lớp học, trang đăng nhập, trang camera. Thu thập video/ảnh từ camera, gửi lên backend để xử lý nhận diện. Hiển thị kết quả nhận diện và thông tin điểm danh theo thời gian thực.
- Backend (Flask – Python): Nhận ảnh được gửi từ frontend. Gọi các module xử lý ảnh, phát hiện khuôn mặt và nhận diện khuôn mặt. Cung cấp API RESTful kết nối frontend – AI – database.
- Module trí tuệ nhân tạo (AI Processing): SCRFD (phát hiện khuôn mặt), ArcFace + iResNet-100 (trích xuất và so khớp vector khuôn mặt), OpenCV, PyTorch, TorchVision để xử lý ảnh.
- MySQL Database: Lưu trữ toàn bộ thông tin hệ thống.



Hình 4: Kiến trúc hệ thống Website

## **Chương 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

### **2.1. Khảo sát yêu cầu thực tế**

#### **2.1.1. Thực trạng công tác điểm danh hiện nay**

Tại nhiều trường học và cơ sở đào tạo, quá trình điểm danh sinh viên vẫn được thực hiện theo phương pháp thủ công như gọi tên, ký tên hoặc quét mã QR. Những phương pháp này tuy đơn giản nhưng tồn tại nhiều hạn chế như:

- Tốn nhiều thời gian, đặc biệt đối với lớp học có số lượng sinh viên đông.
- Dễ xảy ra gian lận khi sinh viên điểm danh hộ hoặc ký tên thay.
- Khó khăn trong việc tổng hợp, thống kê và lưu trữ dữ liệu điểm danh.
- Thiếu tính tự động hóa và không phù hợp với xu hướng chuyển đổi số trong giáo dục.

Trong bối cảnh công nghệ trí tuệ nhân tạo (AI) và thị giác máy tính (Computer Vision) phát triển mạnh mẽ, việc ứng dụng các công nghệ này để tự động hóa công tác điểm danh là hoàn toàn khả thi, giúp nâng cao tính chính xác, giảm thiểu sai sót và tiết kiệm thời gian cho giảng viên.

#### **2.1.2. Mục tiêu khảo sát**

Mục tiêu của quá trình khảo sát là nhằm:

- Nắm bắt thực trạng quy trình điểm danh hiện nay trong các lớp học.
- Xác định nhu cầu, khó khăn và mong muốn của người dùng (giảng viên, quản trị viên).
- Đề xuất giải pháp ứng dụng công nghệ nhận diện khuôn mặt nhằm tối ưu quy trình điểm danh và quản lý dữ liệu sinh viên.

#### **2.1.3. Đối tượng khảo sát**

Quá trình khảo sát được thực hiện với các đối tượng chính sau:

- Giảng viên: Người trực tiếp sử dụng hệ thống để điểm danh, theo dõi và quản lý kết quả học viên.
- Sinh viên: Đối tượng được nhận diện và ghi nhận tham dự buổi học.
- Quản trị viên: Người chịu trách nhiệm quản lý dữ liệu người dùng, lớp học, và hệ thống điểm danh tổng thể.

#### **2.1.4. Kết quả khảo sát**

Sau khi tiến hành khảo sát thực tế tại Khoa Công nghệ thông tin – Trường Đại học Kiến trúc Đà Nẵng, có thể rút ra một số kết luận:

- Phần lớn giảng viên mong muốn có một công cụ điểm danh tự động, giúp tiết kiệm thời gian và đảm bảo tính minh bạch.
- Sinh viên cũng đánh giá cao phương pháp điểm danh bằng khuôn mặt vì nhanh, chính xác, không gây gián đoạn buổi học.
- Nhà trường mong muốn hệ thống có thể quản lý tập trung dữ liệu điểm danh, dễ tra cứu và xuất báo cáo theo lớp hoặc học kỳ.

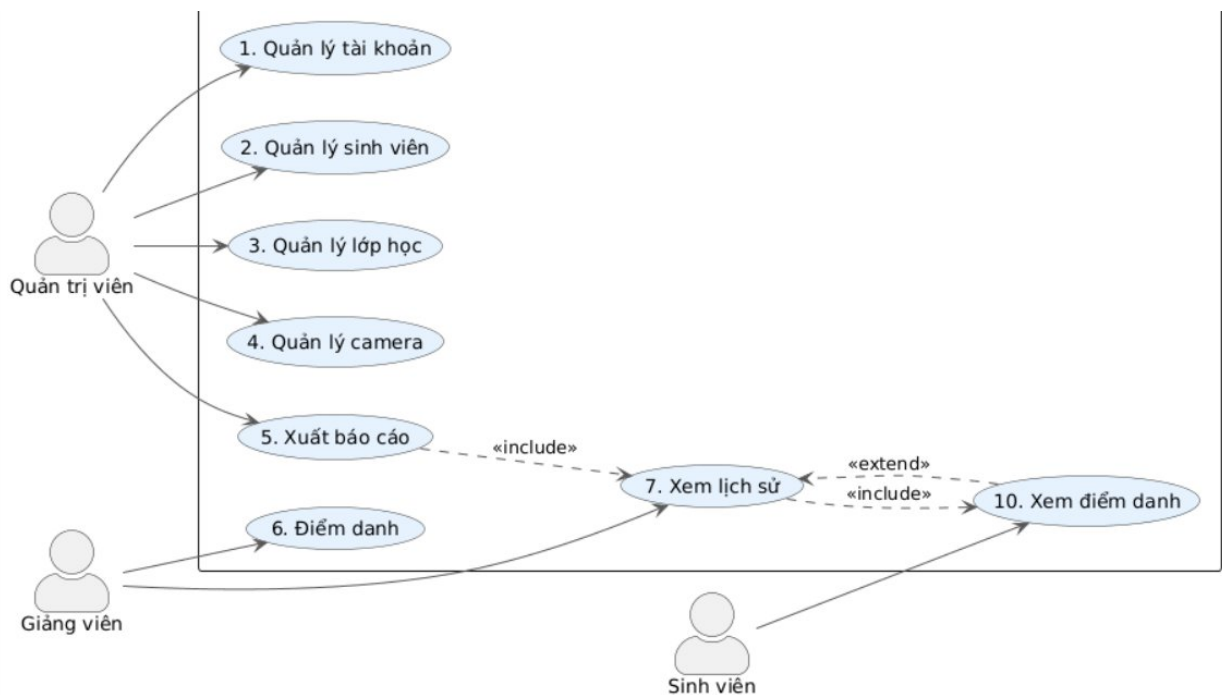
### 2.1.5. Đề xuất hướng giải pháp

Từ kết quả khảo sát, nhóm thực hiện đề xuất xây dựng hệ thống website điểm danh sinh viên bằng nhận diện khuôn mặt với các yêu cầu chính:

- Tự động nhận diện và ghi nhận sinh viên có mặt thông qua camera.
- Quản lý danh sách sinh viên, lớp học và lịch học.
- Lưu trữ dữ liệu điểm danh tập trung trên máy chủ.
- Cung cấp giao diện thân thiện, dễ sử dụng cho giảng viên và quản trị viên.
- Ứng dụng công nghệ AI và OpenCV để xử lý hình ảnh và nhận diện khuôn mặt với độ chính xác cao.

## 2.2. Phân tích hệ thống

### 2.2.1. Mô hình UseCase



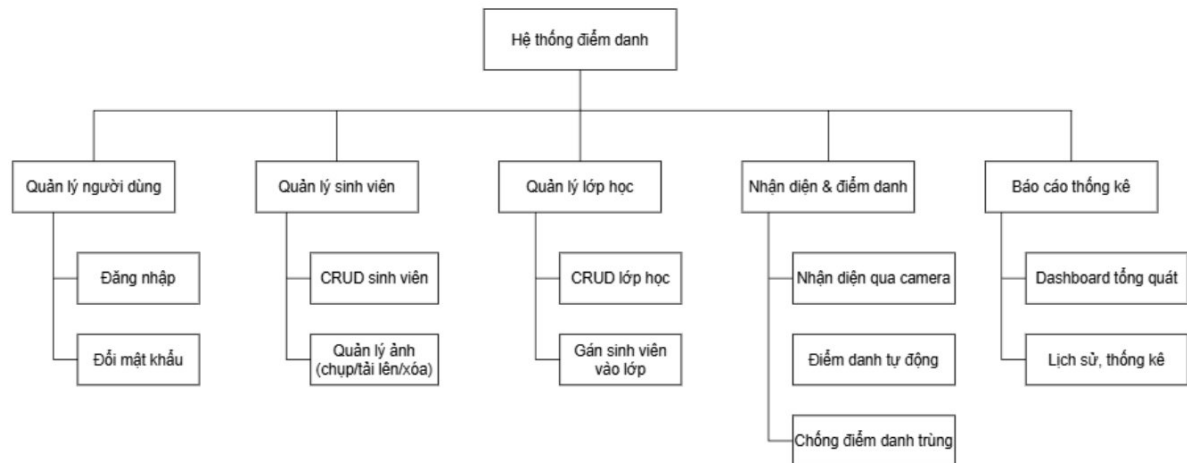
Hình 5: Mô hình UseCase tổng quát

| STT | Tên Use Case | Tác nhân (Actor) | Mô tả chức năng |
|-----|--------------|------------------|-----------------|
|-----|--------------|------------------|-----------------|

|   |                   |               |                                                                                                                                                       |
|---|-------------------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Quản lý tài khoản | Quản trị viên | Đăng nhập, đăng xuất hệ thống. Phân quyền truy cập cho giảng viên và các tài khoản quản trị khác.                                                     |
| 2 | Quản lý sinh viên | Quản trị viên | Thêm mới, cập nhật, xóa thông tin hồ sơ sinh viên. Quan trọng nhất là chức năng nạp và quản lý dữ liệu hình ảnh khuôn mặt để huấn luyện cho AI.       |
| 3 | Quản lý lớp học   | Quản trị viên | Tạo lập các lớp học hành chính hoặc lớp học phân, gán danh sách sinh viên vào các lớp tương ứng.                                                      |
| 4 | Quản lý camera    | Quản trị viên | Cấu hình kết nối với thiết bị camera (IP/RTSP) hoặc Webcam. Thiết lập các thông số kỹ thuật đầu vào cho module nhận diện.                             |
| 5 | Xuất báo cáo      | Quản trị viên | Trích xuất dữ liệu điểm danh từ hệ thống ra các định dạng file lưu trữ (Excel, PDF) để phục vụ công tác thống kê, báo cáo định kỳ.                    |
| 6 | Điểm danh         | Giảng viên    | Kích hoạt hệ thống camera tại lớp học để tự động nhận diện khuôn mặt sinh viên và ghi nhận trạng thái "Có mặt" vào cơ sở dữ liệu theo thời gian thực. |
| 7 | Xem lịch sử       | Giảng viên    | Xem lại nhật ký ra/vào và danh sách điểm danh của các lớp học mà giảng viên phụ trách để theo dõi tình hình chuyên cần.                               |
| 8 | Xem điểm danh     | Sinh viên     | Sinh viên đăng nhập vào hệ thống để tự kiểm tra trạng thái điểm danh của cá nhân mình trong các buổi học.                                             |

Bảng 1: Mô tả mô hình UseCase

### 2.2.2. Sơ đồ chức năng



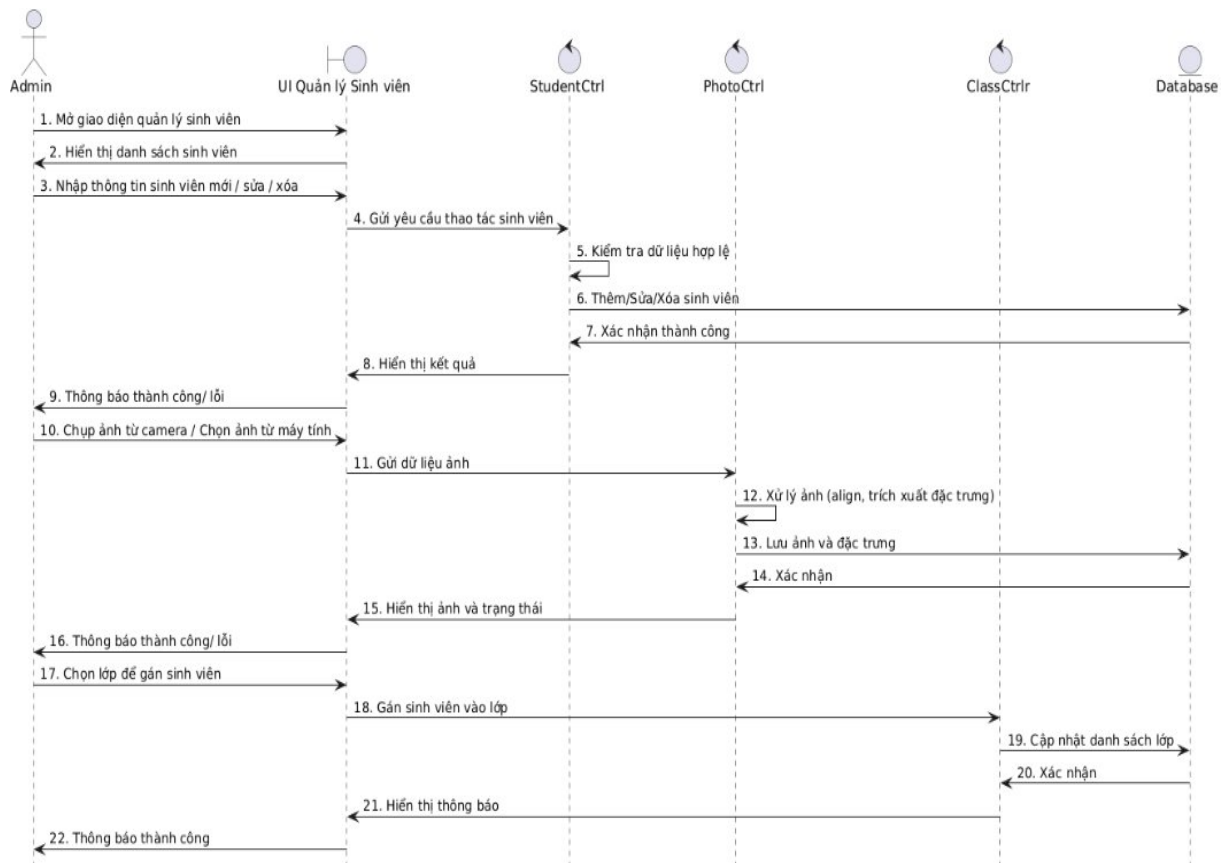
Hình 6: Sơ đồ chức năng

Đặc tả các module chính:

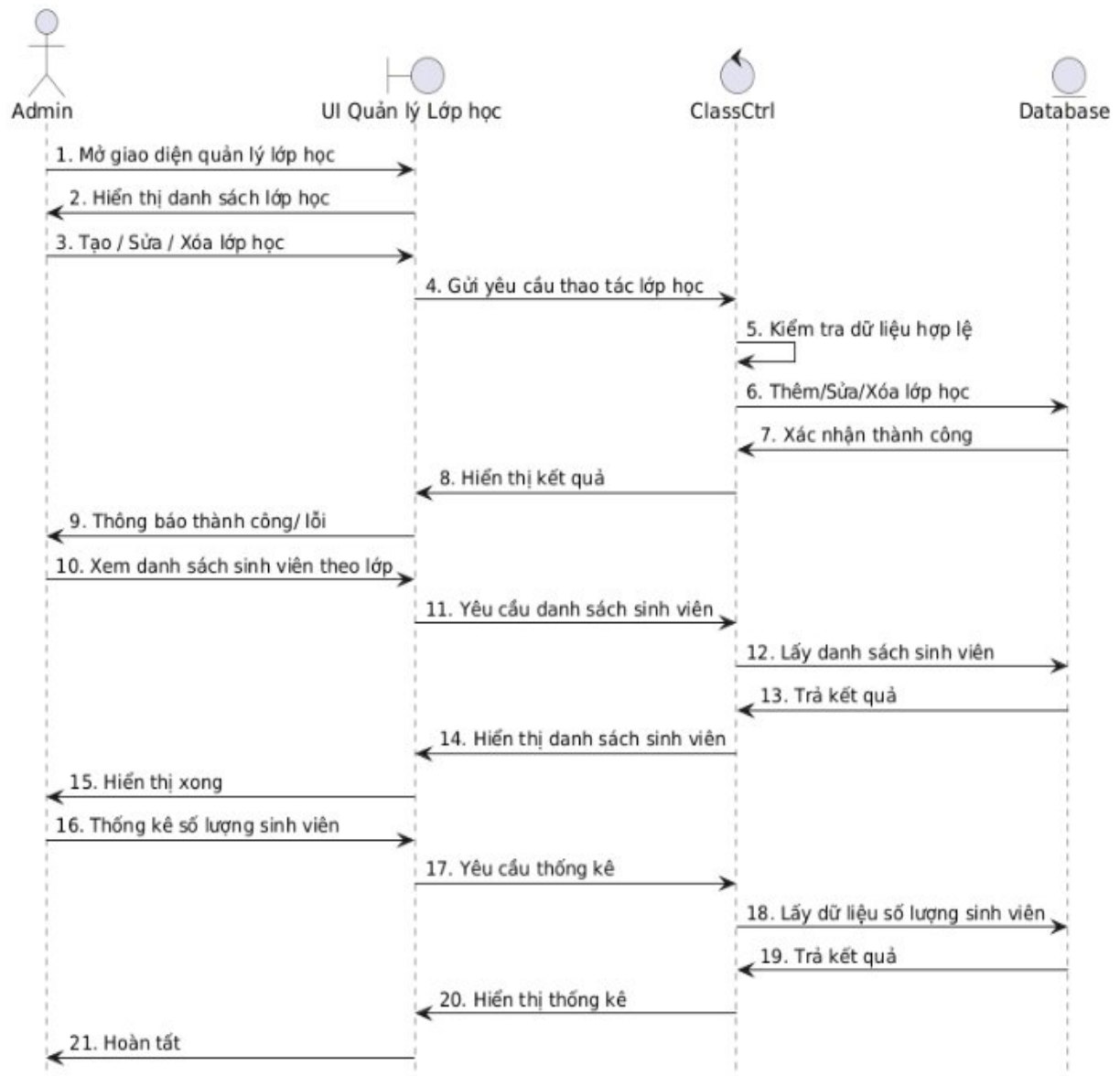
- Quản lý người dùng: Đảm bảo chỉ những người có quyền (Admin) mới có thể truy cập và sử dụng hệ thống.
- Quản lý sinh viên: Lưu trữ và quản lý toàn bộ thông tin sinh viên phục vụ cho việc nhận diện và điểm danh.
- Quản lý lớp học: Tổ chức các lớp học – nền tảng để hệ thống xác định sinh viên thuộc buổi điểm danh nào.
- Nhận diện & Điểm danh: Thực hiện chức năng cốt lõi của hệ thống – nhận diện sinh viên và ghi nhận việc điểm danh.
- Báo cáo & Thống kê: Hỗ trợ Admin theo dõi, đánh giá và quản lý tình hình học tập thông qua dữ liệu điểm danh.



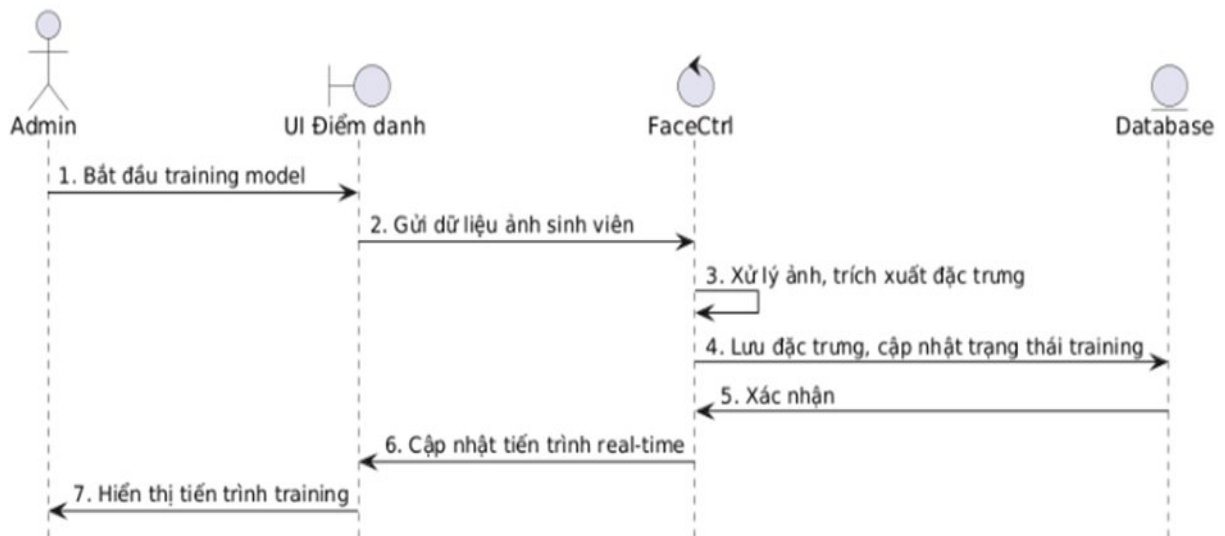
## 2.2.4. Biểu đồ tuần tự



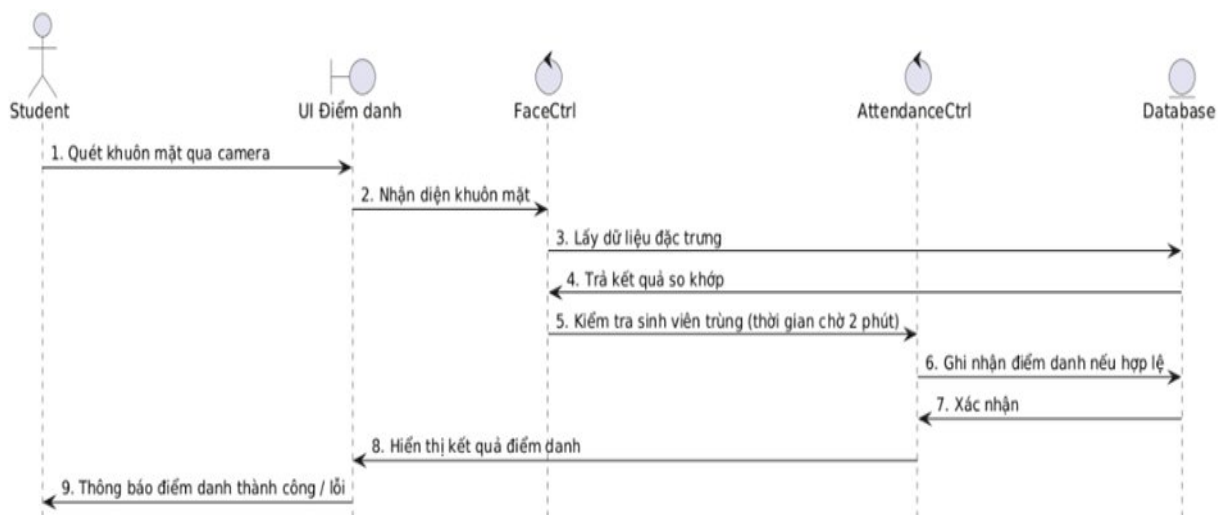
Hình 9: Biểu đồ tuần tự quản lý sinh viên



Hình 10: Biểu đồ tuần tự quản lý lớp học

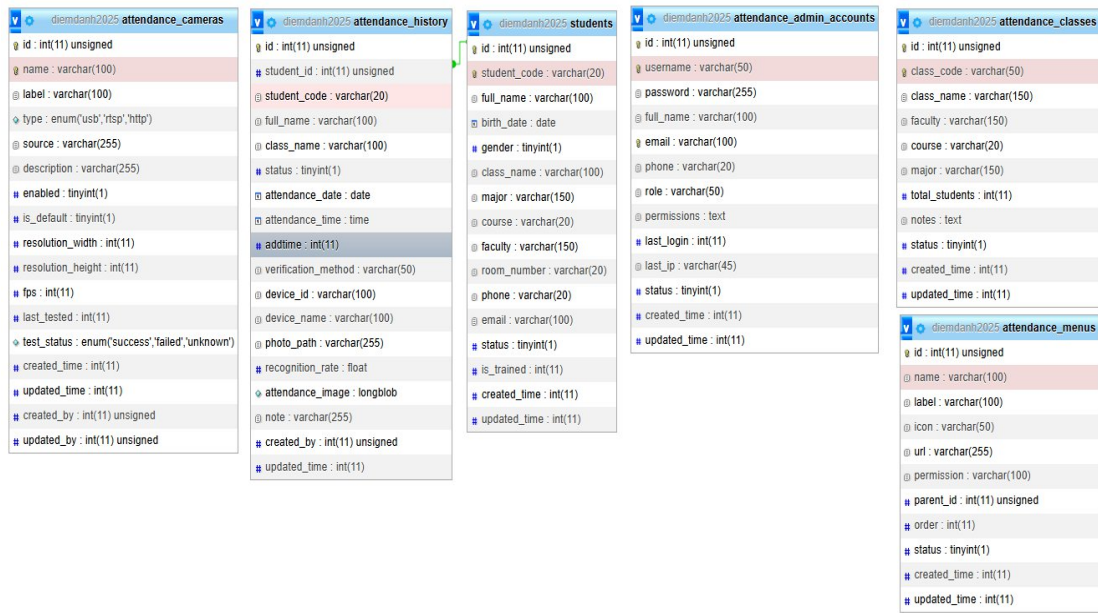


Hình 11: Biểu đồ tuần tự Training model



Hình 12: Biểu đồ tuần tự điểm danh

### 2.2.5. Thiết kế cơ sở dữ liệu



Hình 13: Mô hình quan hệ dữ liệu của hệ thống

| Tên trường   | Kiểu dữ liệu                      | Mô tả chức năng                                                        |
|--------------|-----------------------------------|------------------------------------------------------------------------|
| id           | int(11) unsigned – AUTO INCREMENT | Khóa chính, định danh duy nhất cho mỗi tài khoản admin.                |
| username     | varchar(50)                       | Tên đăng nhập vào hệ thống.                                            |
| password     | varchar(255)                      | Mật khẩu đã mã hóa.                                                    |
| full_name    | varchar(100)                      | Họ tên quản trị viên.                                                  |
| email        | varchar(100)                      | Email dùng để nhận thông báo hoặc khôi phục mật khẩu.                  |
| phone        | varchar(20)                       | Số điện thoại liên hệ.                                                 |
| role         | varchar(50)                       | Vai trò của tài khoản.                                                 |
| permissions  | text                              | Danh sách quyền hạn dạng JSON, cho phép tùy chỉnh chi tiết từng quyền. |
| last_login   | int(11)                           | Thời gian đăng nhập gần nhất (timestamp).                              |
| last_ip      | varchar(45)                       | Địa chỉ IP lần đăng nhập cuối.                                         |
| status       | tinyint(1)                        | Trạng thái tài khoản: 0 = Khóa, 1 = Hoạt động                          |
| created_time | int(11)                           | Thời gian tạo tài khoản (timestamp).                                   |
| updated_time | int(11)                           | Thời gian cập nhật gần nhất.                                           |

Bảng 2: Mô tả chi tiết bảng attendance\_admin\_accounts

| Tên trường        | Kiểu dữ liệu                       | Mô tả chức năng                                             |
|-------------------|------------------------------------|-------------------------------------------------------------|
| id                | int(11) unsigned                   | ID duy nhất của camera (khóa chính, tự tăng).               |
| name              | varchar(100)                       | Tên định danh camera (VD: webcam_0, ip_camera_1).           |
| label             | varchar(100)                       | Nhãn hiển thị thân thiện trên giao diện.                    |
| type              | enum('usb','rtsp','http')          | Loại camera: USB, RTSP hoặc HTTP.                           |
| source            | varchar(255)                       | Nguồn camera: index USB hoặc URL RTSP/HTTP.                 |
| description       | varchar(255)                       | Mô tả vị trí/chức năng của camera.                          |
| enabled           | tinyint(1)                         | Trạng thái hoạt động: 1 = bật, 0 = tắt.                     |
| is_default        | tinyint(1)                         | Camera mặc định của hệ thống (1 = có).                      |
| resolution_width  | int(11)                            | Độ rộng video (px).                                         |
| resolution_height | int(11)                            | Độ cao video (px).                                          |
| fps               | int(11)                            | Số khung hình/giây (Frames Per Second).                     |
| last_tested       | int(11)                            | Thời gian test kết nối gần nhất (timestamp).                |
| test_status       | enum('success','failed','unknown') | Trạng thái test kết nối: thành công/thất bại/chưa xác định. |
| created_time      | int(11)                            | Thời gian tạo bản ghi.                                      |
| updated_time      | int(11)                            | Thời gian cập nhật bản ghi.                                 |
| created_by        | int(11) unsigned                   | ID người tạo bản ghi.                                       |
| updated_by        | int(11) unsigned                   | ID người cập nhật bản ghi.                                  |

Bảng 3: Mô tả chi tiết bảng *attendance\_cameras*

| Tên trường     | Kiểu dữ liệu                | Mô tả chức năng                                                                                     |
|----------------|-----------------------------|-----------------------------------------------------------------------------------------------------|
| id             | INT(11) -<br>AUTO_INCREMENT | Khóa chính (Primary Key), tự động tăng. Dùng để định danh duy nhất cho từng lớp học trong hệ thống. |
| class_code     | VARCHAR(50)                 | Mã lớp hành chính (Ví dụ: 21CT5).                                                                   |
| class_name     | VARCHAR(150)                | Tên đầy đủ của lớp học.                                                                             |
| faculty        | VARCHAR(150)                | Tên Khoa quản lý lớp học này (Ví dụ: Khoa Công nghệ thông tin).                                     |
| course         | VARCHAR(20)                 | Khóa học của sinh viên                                                                              |
| major          | VARCHAR(150)                | Chuyên ngành đào tạo cụ thể của lớp.                                                                |
| total_students | INT(11)                     | Lưu tổng số lượng sinh viên hiện có trong lớp.                                                      |
| notes          | TEXT                        | Trường dùng để ghi chú thêm thông tin về lớp học.                                                   |
| status         | TINYINT(1)                  | Trạng thái hoạt động của lớp học.                                                                   |
| created_time   | INT(11)                     | Thời gian tạo lớp học                                                                               |
| updated_time   | INT(11)                     | Thời gian cập nhật thông tin lớp học gần nhất.                                                      |

Bảng 4: Mô tả chi tiết bảng *attendance\_classes*

| Tên trường          | Kiểu dữ liệu                | Mô tả chức năng                                                                    |
|---------------------|-----------------------------|------------------------------------------------------------------------------------|
| id                  | int(11) -<br>AUTO_INCREMENT | Khóa chính (Primary Key), tự động tăng. Định danh duy nhất cho mỗi lượt điểm danh. |
| student_id          | int(11)                     | Dùng để xác định sinh viên nào đang điểm danh.                                     |
| student_code        | varchar(20)                 | Mã sinh viên.                                                                      |
| full_name           | varchar(100)                | Họ tên sinh viên tại thời điểm điểm danh.                                          |
| class_name          | varchar(100)                | Tên lớp học của sinh viên.                                                         |
| status              | tinyint(1)                  | Trạng thái điểm danh.                                                              |
| attendance_date     | date                        | Ngày thực hiện điểm danh.                                                          |
| attendance_time     | time                        | Giờ thực hiện điểm danh                                                            |
| addtime             | int(11)                     | Thời gian ghi nhận bản ghi.                                                        |
| verification_method | varchar(50)                 | Phương thức xác thực.                                                              |
| device_id           | varchar(100)                | Mã định danh của thiết bị thực hiện điểm danh.                                     |
| device_name         | varchar(100)                | Tên của thiết bị điểm danh.                                                        |
| photo_path          | varchar(255)                | Đường dẫn lưu trữ file ảnh chụp bằng chứng khi điểm danh trên server.              |
| recognition_rate    | float                       | Độ chính xác.                                                                      |
| attendance_image    | longblob                    | Lưu trữ dữ liệu ảnh.                                                               |
| note                | varchar(255)                | Ghi chú thêm.                                                                      |
| created_by          | int(11)                     | ID của người tạo bản ghi này.                                                      |
| updated_time        | int(11)                     | Thời gian cập nhật/chỉnh sửa bản ghi gần nhất.                                     |

Bảng 5: Mô tả chi tiết attendance\_history

| Tên trường   | Kiểu dữ liệu                | Mô tả chức năng                                                                               |
|--------------|-----------------------------|-----------------------------------------------------------------------------------------------|
| id           | int(11) -<br>AUTO_INCREMENT | Khóa chính (Primary Key), tự động tăng.<br>Định danh duy nhất cho từng mục menu.              |
| name         | varchar(100)                | Tên định danh của menu item trong hệ thống.                                                   |
| label        | varchar(100)                | Nhãn hiển thị của menu trên giao diện người dùng.                                             |
| icon         | varchar(50)                 | Lưu tên class của icon để hiển thị biểu tượng minh họa cạnh tên menu.                         |
| url          | varchar(255)                | Đường dẫn (route) hoặc hash mà trình duyệt sẽ chuyển hướng tới khi người dùng click vào menu. |
| permission   | varchar(100)                | Mã quyền hạn bắt buộc để nhìn thấy menu này. Giúp ẩn các menu quản trị khỏi user thường.      |
| parent_id    | int(11)                     | ID của menu cha.                                                                              |
| order        | int(11)                     | Số thứ tự sắp xếp hiển thị của các menu.                                                      |
| status       | tinyint(1)                  | Trạng thái hiển thị của menu.                                                                 |
| created_time | int(11)                     | Thời gian tạo menu.                                                                           |
| updated_time | int(11)                     | Thời gian cập nhật thông tin menu gần nhất.                                                   |

*Bảng 6: Mô tả chi tiết bảng attendance\_menus*

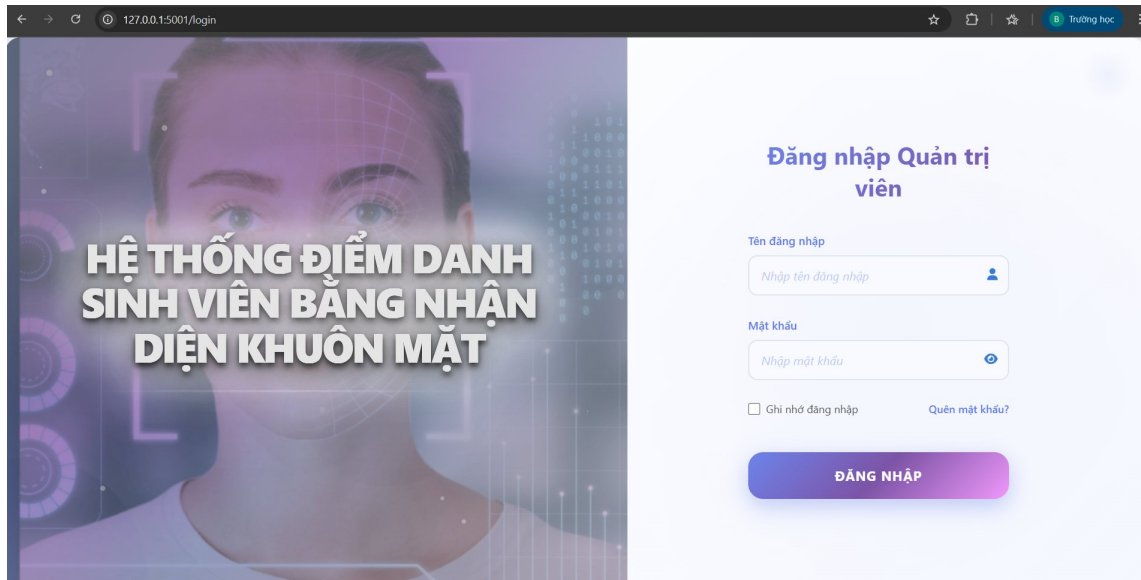
| Tên trường   | Kiểu dữ liệu                | Mô tả chức năng                                                                               |
|--------------|-----------------------------|-----------------------------------------------------------------------------------------------|
| id           | int(11) -<br>AUTO_INCREMENT | Khóa chính (Primary Key), tự động tăng. Định danh duy nhất cho từng sinh viên trong hệ thống. |
| student_code | varchar(20)                 | Mã số sinh viên                                                                               |
| full_name    | varchar(100)                | Họ và tên đầy đủ của sinh viên.                                                               |
| birth_date   | date                        | Ngày tháng năm sinh.                                                                          |
| gender       | tinyint(1)                  | Giới tính của sinh viên.                                                                      |
| class_name   | varchar(100)                | Tên lớp hành chính mà sinh viên đang theo học.                                                |
| major        | varchar(150)                | Chuyên ngành đào tạo của sinh viên.                                                           |
| course       | varchar(20)                 | Khóa học của sinh viên.                                                                       |
| faculty      | varchar(150)                | Tên Khoa quản lý sinh viên.                                                                   |
| phone        | varchar(20)                 | Số điện thoại liên lạc của sinh viên.                                                         |
| email        | varchar(100)                | Địa chỉ Email cá nhân hoặc Email do nhà trường cấp.                                           |
| status       | tinyint(1)                  | Trạng thái hoạt động.                                                                         |
| is_trained   | int(11)                     | Cờ đánh dấu trạng thái huấn luyện khuôn mặt.                                                  |
| created_time | int(11)                     | Thời gian tạo hồ sơ sinh viên.                                                                |
| updated_time | int(11)                     | Thời gian cập nhật hồ sơ gần nhất.                                                            |

*Bảng 7: Mô tả chi tiết bảng students*

## 2.2.6. Giao diện người dùng

### 2.2.6.1. Đăng nhập

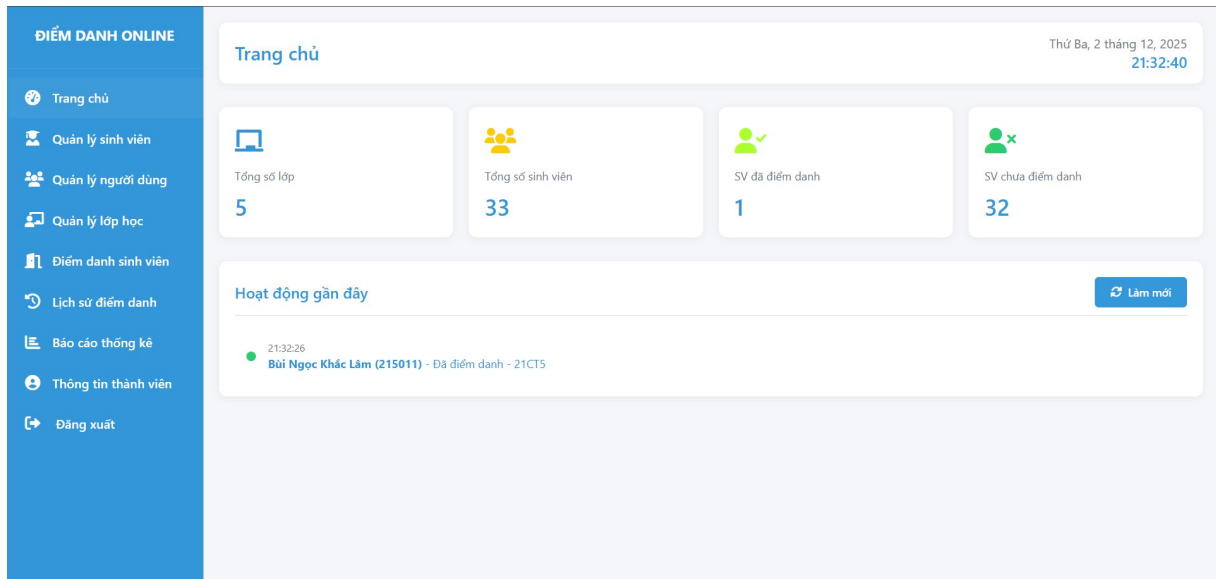
Giao diện đăng nhập được thiết kế đơn giản, hiện đại, gồm hình nền minh họa công nghệ nhận diện khuôn mặt và khung đăng nhập dành cho quản trị viên với các trường thông tin cơ bản và nút đăng nhập rõ ràng, dễ sử dụng.



Hình 14: Giao diện trang đăng nhập

### 2.2.6.2. Trang chủ

Giao diện trang chủ được thiết kế theo phong cách Dashboard (Bảng điều khiển). Mục tiêu là cung cấp cái nhìn tổng quan nhanh chóng về trạng thái hệ thống. Khu vực Chính hiển thị các chỉ số thống kê quan trọng (Key Metrics) như tổng số lớp, tổng số sinh viên, số lượng sinh viên đã điểm danh, chưa điểm danh và thông tin sinh viên điểm danh gần đây.

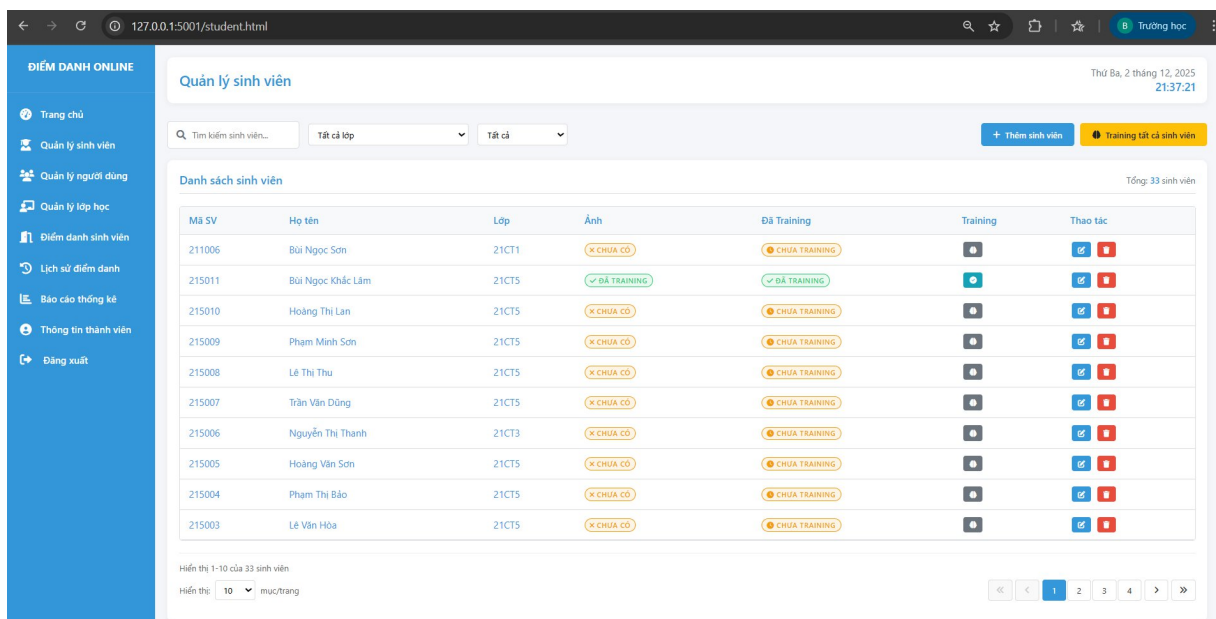


Hình 15: Giao diện trang chủ

### 2.2.6.3. Quản lý sinh viên

Module quản lý sinh viên đóng vai trò là trung tâm dữ liệu để quản lý thông tin và hình ảnh khuôn mặt của sinh viên. Chức năng chính bao gồm:

- CRUD Dữ liệu: Thực hiện các thao tác Thêm, Chỉnh sửa, Xóa thông tin cơ bản của sinh viên (Mã sinh viên, Tên, Lớp).
- Quản lý Hình ảnh: Cho phép Tải lên/Cập nhật ảnh khuôn mặt của sinh viên, là dữ liệu đầu vào bắt buộc cho công nghệ nhận diện.
- Tra cứu: Cung cấp chức năng Tìm kiếm và Lọc theo Mã sinh viên hoặc Lớp học để dễ dàng thao tác với danh sách.

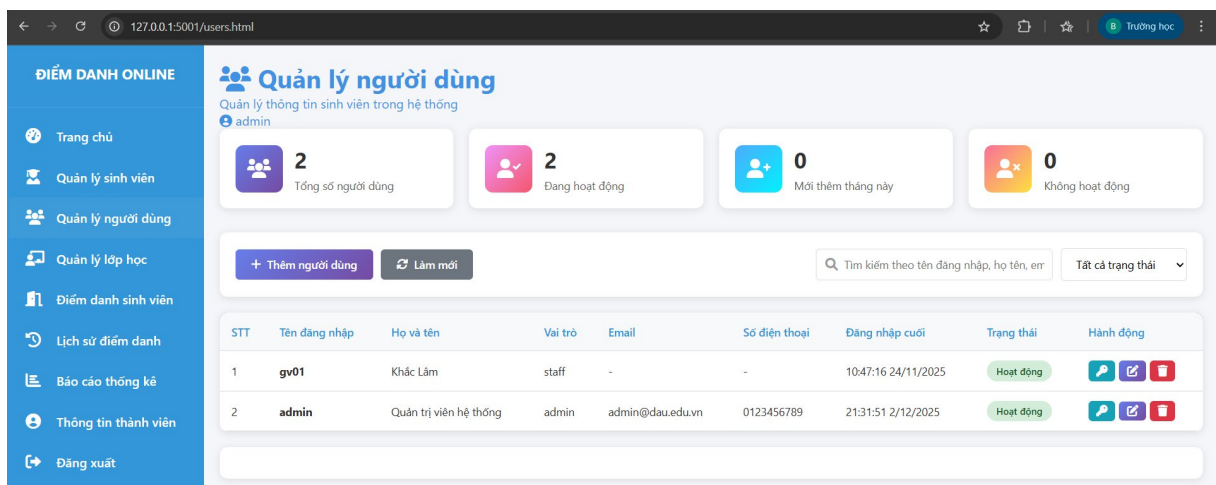


Hình 16: Giao diện quản lý sinh viên

#### 2.2.6.4. Quản lý người dùng

Module quản lý người dùng có chức năng quản lý các tài khoản và phân quyền truy cập hệ thống.

- Phân quyền: Xác định các vai trò chính (ví dụ: Quản trị viên và Giảng viên), mỗi vai trò có phạm vi quyền hạn khác nhau để đảm bảo bảo mật và tính toàn vẹn của dữ liệu.
- Chức năng: Cho phép Thêm, Chỉnh sửa, Vô hiệu hóa tài khoản, và Thiết lập lại mật khẩu.

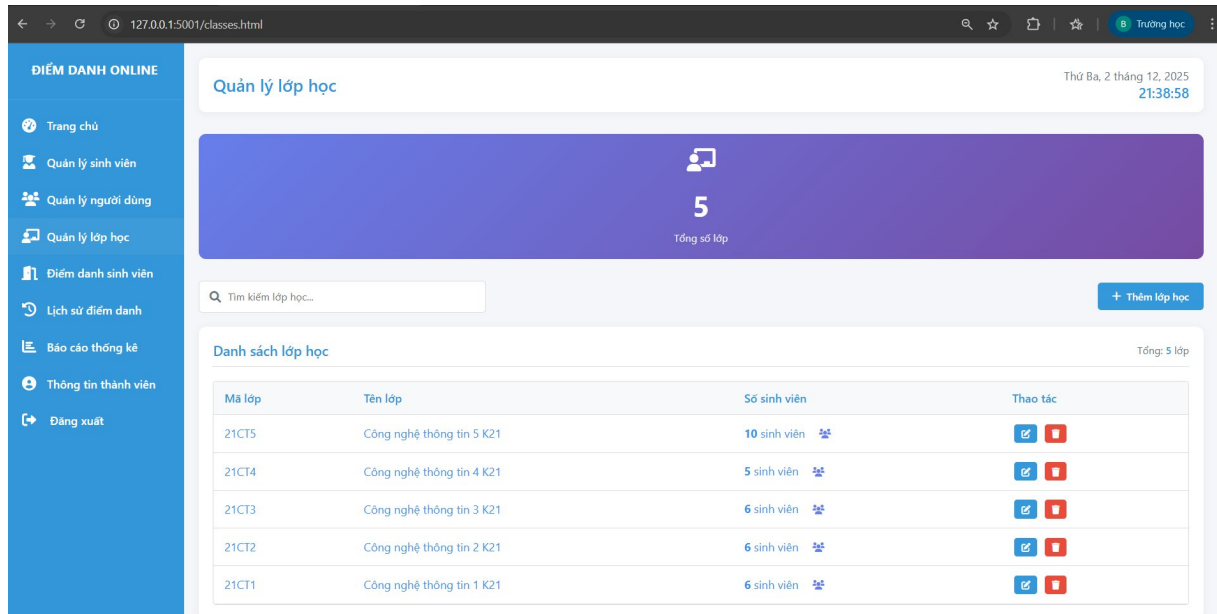


Hình 17: Giao diện quản lý người dùng

#### 2.2.6.5. Quản lý lớp học

Module Quản lý Lớp học cung cấp giao diện để người dùng (Giảng viên/Admin) tổ chức và quản lý các nhóm sinh viên học tập.

Tạo/Sửa/Xóa Lớp (CRUD): Cho phép tạo lớp học mới, chỉnh sửa thông tin lớp (Tên lớp, Mã lớp, Giảng viên phụ trách) hoặc xóa lớp.

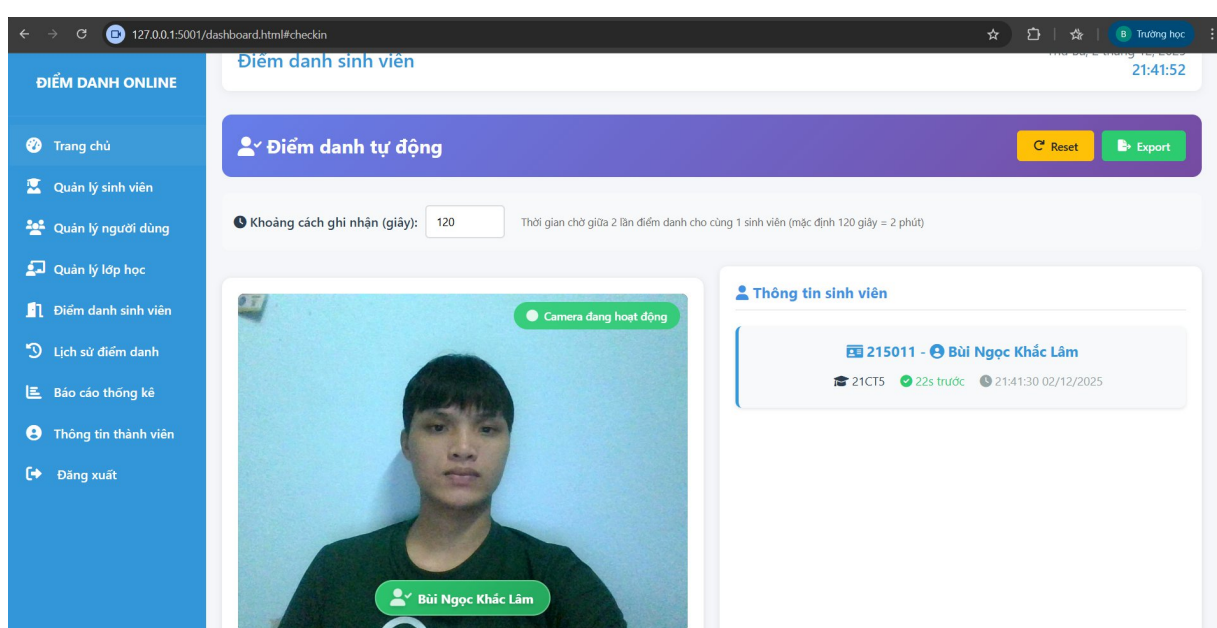


Hình 18: Giao diện quản lý lớp học

#### 2.2.6.6. Điểm danh sinh viên

Đây là chức năng cốt lõi, thực hiện việc điểm danh tự động và theo thời gian thực (real-time) bằng cách ứng dụng công nghệ Nhận diện Khuôn mặt (Face Recognition).

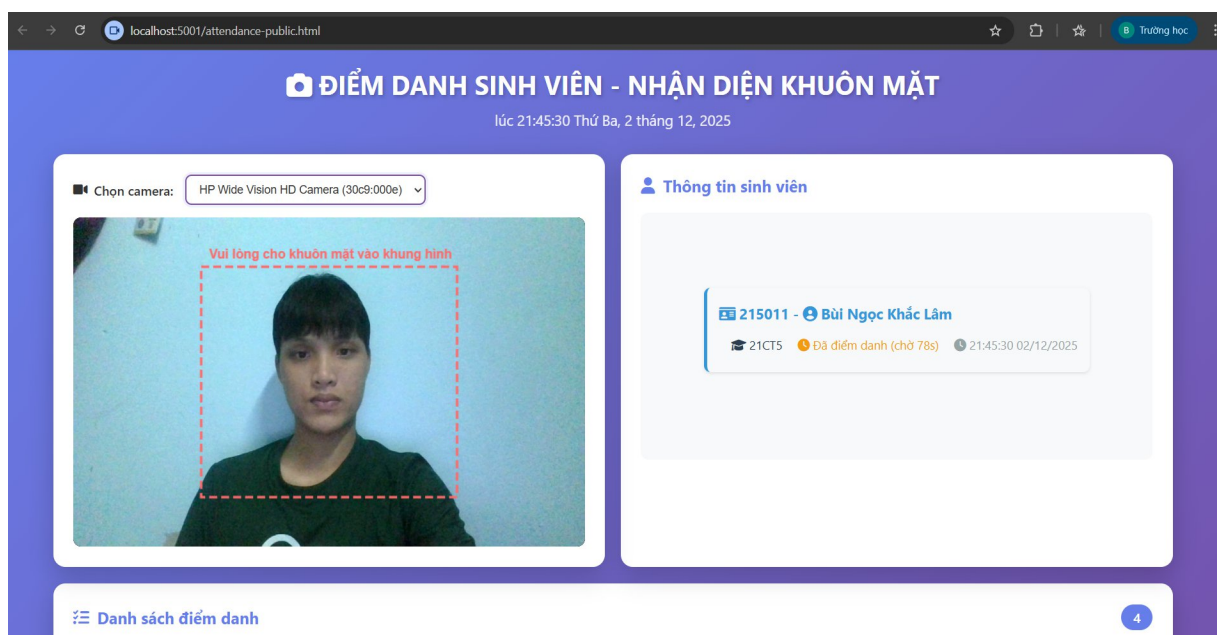
- Quy trình: Giảng viên kích hoạt camera; hệ thống xác minh danh tính sinh viên có mặt trong khung hình bằng cách so sánh với cơ sở dữ liệu đã huấn luyện.
- Kết quả: Sinh viên được nhận diện thành công sẽ tự động được ghi nhận trạng thái Có mặt (Present) vào lịch sử điểm danh của buổi học.
- Hỗ trợ: Giao diện cho phép giảng viên theo dõi trạng thái điểm danh.



Hình 19: Giao diện điểm danh sinh viên trên hệ thống quản trị viên

Ngoài ra còn có giao diện điểm danh công khai được thiết kế để sinh viên tự thực hiện điểm danh mà không cần đăng nhập.

- Tính năng: Màn hình chỉ tập trung vào luồng video từ camera. Sinh viên chỉ cần nhìn vào camera để kích hoạt quá trình nhận diện khuôn mặt.
- Phản hồi: Sau khi nhận diện thành công, hệ thống hiển thị thông báo xác nhận nhanh (ví dụ: "Điểm danh thành công!") cùng với tên sinh viên, sau đó tự động trở về trạng thái chờ, đảm bảo tốc độ và sự tiện lợi tối đa.

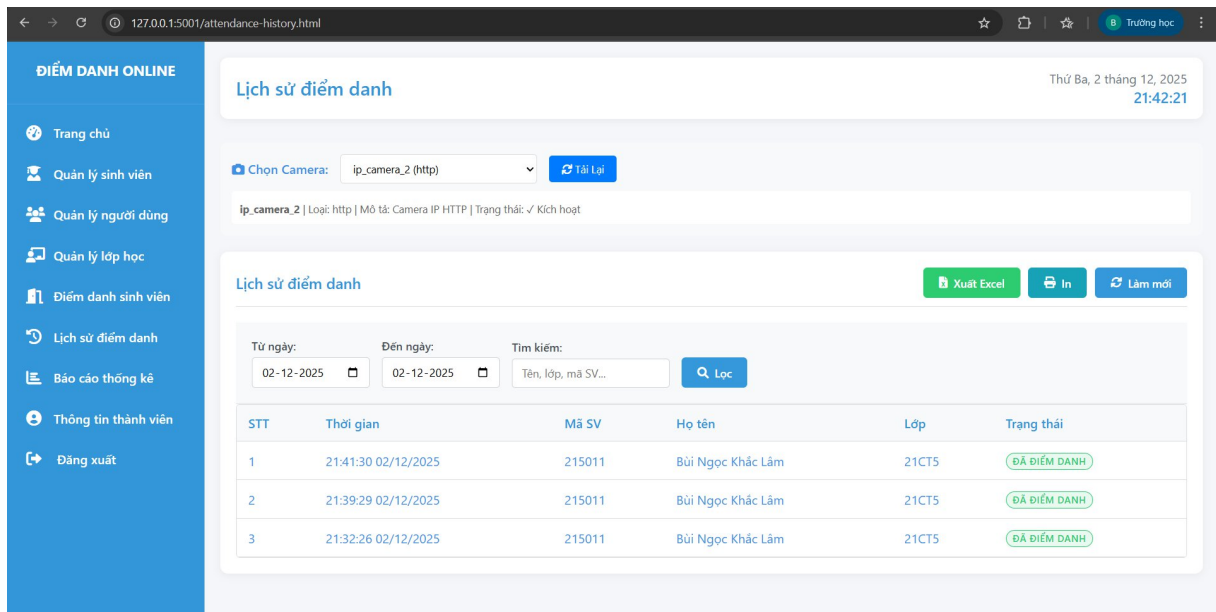


Hình 20: Giao diện điểm danh sinh viên công khai

#### 2.2.6.7. Lịch sử điểm danh

Module này có chức năng quản lý, lưu trữ và truy xuất toàn bộ dữ liệu điểm danh đã được ghi nhận.

- Xem chi tiết: Cho phép xem lại kết quả điểm danh chi tiết của từng buổi học và từng sinh viên.
- Truy vấn: Hỗ trợ Tìm kiếm và Lọc dữ liệu theo Lớp học, Ngày tháng, và Trạng thái (Có mặt/Vắng mặt).
- Xuất dữ liệu: Cung cấp tính năng Xuất báo cáo ra các định dạng phổ biến như Excel/CSV để phục vụ cho các công tác báo cáo và lưu trữ ngoại tuyến.

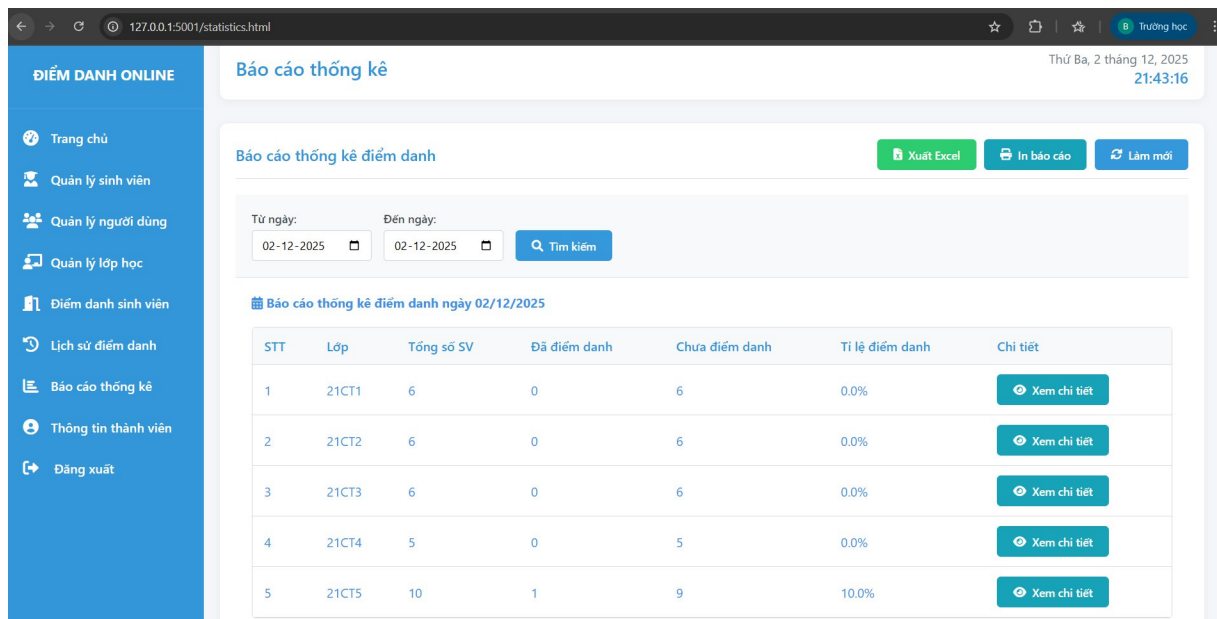


Hình 21: Giao diện lịch sử điểm danh

#### 2.2.6.8. Báo cáo thống kê

Module báo cáo thống kê đóng vai trò quan trọng trong việc phân tích và trực quan hóa dữ liệu điểm danh đã thu thập.

Chức năng: Chuyển đổi dữ liệu lịch sử điểm danh thành các báo cáo dễ hiểu thông qua Biểu đồ và Bảng tổng hợp. Theo dõi được số lượng sinh viên đã và chưa điểm danh theo từng lớp.



Hình 22: Giao diện báo cáo thống kê

## Chương 3: XÂY DỰNG VÀ TRIỂN KHAI HỆ THỐNG

### 3.1.1. Môi trường và công cụ phát triển

#### 3.1.2. Công cụ phát triển

Hệ thống được xây dựng dựa trên sự hỗ trợ của các công cụ và phần mềm sau:

- Môi trường tích hợp (IDE/Editor): Visual Studio Code, sử dụng kèm các extension hỗ trợ như Python, Pylance, Pylint để tối ưu hóa việc viết mã. PyCharm (Phiên bản Professional) hỗ trợ mạnh mẽ cho việc debug và phát triển ứng dụng Web Python.
- Quản lý phiên bản (Version Control): Sử dụng Git kết hợp với các nền tảng lưu trữ như GitHub để quản lý mã nguồn.
- Cơ sở dữ liệu: MySQL (Hệ quản trị cơ sở dữ liệu chính), phpMyAdmin (Công cụ quản lý).
- Công cụ Kiểm thử (Testing): Postman, sử dụng để kiểm thử các API endpoint.

Cấu trúc tổ chức hệ thống

face-recognition/

|                     |                          |
|---------------------|--------------------------|
| — database/         | # Cơ sở dữ liệu          |
| — datasets/         | # Dữ liệu khuôn mặt      |
| — data/             | # Ảnh khuôn mặt gốc      |
| — face_features/    | # Đặc trưng khuôn mặt    |
| — static/           | # Tài nguyên tĩnh        |
| — face_alignment/   | # Căn chỉnh khuôn mặt    |
| — face_detection/   | # Phát hiện khuôn mặt    |
| — face_recognition/ | # Nhận dạng khuôn mặt    |
| — api.py            | # Ứng dụng Flask chính   |
| — camera_manager.py | # Quản lý camera         |
| — db_config.py      | # Cấu hình cơ sở dữ liệu |
| — permissions.py    | # Quản lý phân quyền     |
| — requirements.txt  | # Thư viện cần thiết     |

#### 3.1.3. Quy trình triển khai hệ thống

Bước 1: Thiết lập môi trường ảo (Virtual Environment) Việc sử dụng môi trường ảo giúp cô lập các thư viện của dự án, tránh xung đột phiên bản.

# Tạo môi trường ảo

python -m venv venv

# Kích hoạt môi trường ảo (Windows)

.\venv\Scripts\activate

Bước 2: Cài đặt các thư viện phụ thuộc Cài đặt toàn bộ các thư viện cần thiết từ file requirements.txt: `pip install -r requirements.txt`

Bước 3: Cấu hình Cơ sở dữ liệu

- Cài đặt MySQL Server.
- Cập nhật thông tin kết nối (Host, User, Password, Database Name) trong file db\_config.py để khớp với môi trường local.

Bước 4: Khởi chạy ứng dụng Sau khi hoàn tất cấu hình, thực hiện lệnh sau để bắt đầu server: `python api.py`

Bước 5: Mở trình duyệt và truy cập: <http://localhost:5001>

### 3.2. Xây dựng backend và API quản lý lớp học, sinh viên, điểm danh

#### 3.2.1. Mục tiêu

Phần backend được xây dựng nhằm cung cấp hệ thống API RESTful để phục vụ cho các chức năng quản lý của hệ thống điểm danh bằng nhận diện khuôn mặt. Các mục tiêu chính gồm:

- Quản lý dữ liệu lớp học, sinh viên, tài khoản người dùng.
- Tạo và quản lý buổi học (session).
- Thực hiện điểm danh tự động thông qua mô-đun nhận diện khuôn mặt.
- Lưu trữ và cập nhật kết quả điểm danh theo thời gian thực.
- Cung cấp API thống kê, báo cáo chuyên cần.
- Đảm bảo bảo mật, phân quyền, tính toàn vẹn dữ liệu.

Kiến trúc tổng quát:

- API Server (Flask hoặc FastAPI)
- Mô-đun AI
- Database MySQL

#### 3.2.2. Một số đoạn mã minh họa

##### 3.2.2.1. API quản lý lớp học

```
@app.route('/api/classes', methods=['GET'])
def get_classes():
    """Lấy danh sách lớp học"""
    try:
        connection = connect_database()
        if not connection:
            return jsonify({
                'success': False,
                'message': 'Lỗi kết nối database'
            }), 500

        cursor = connection.cursor(dictionary=True)

        page = int(request.args.get('page', 1))
        limit = int(request.args.get('limit', 10))
        search = request.args.get('search', '')

        offset = (page - 1) * limit

        where_clause = "WHERE status = 1"
        params = []
        if search:
            where_clause += " AND (class_code LIKE %s OR class_name LIKE %s OR faculty LIKE %s)"
            search_param = f"%{search}%"
            params.extend([search_param, search_param, search_param])
        # Đếm tổng số bản ghi
        count_query = f"SELECT COUNT(*) as total FROM attendance_classes {where_clause}"
        cursor.execute(count_query, params)
        total = cursor.fetchone()['total']

        # Lấy dữ liệu với phân trang
        query = f"""
        SELECT id, class_code, class_name, created_time
        FROM attendance_classes
        {where_clause}
        ORDER BY created_time DESC
        LIMIT %s OFFSET %s
        """
        params.extend([limit, offset])
        cursor.execute(query, params)
        classes = cursor.fetchall()
        # Format dữ liệu và đếm số sinh viên trong mỗi lớp
        for class_item in classes:
            if class_item['created_time']:
                class_item['created_time'] = datetime.fromtimestamp(class_item['created_time']).strftime('%Y-%m-%d %H:%M:%S')
            # Đếm số sinh viên trong lớp (so sánh bằng class_code)
            count_query = "SELECT COUNT(*) as student_count FROM students WHERE class_name = %s AND status = 1"
            cursor.execute(count_query, (class_item['class_code'],))
            count_result = cursor.fetchone()
            class_item['student_count'] = count_result['student_count'] if count_result else 0

        return jsonify({
            'success': True,
            'data': {
                'classes': classes,
                'pagination': {
                    'page': page,
                    'limit': limit,
                    'total': total,
                    'total_pages': (total + limit - 1) // limit
                }
            }
        })

    finally:
        cursor.close()
        connection.close()

    except Exception as e:
        return jsonify({
            'success': False,
            'message': 'Lỗi server nội bộ'
        }), 500
```

Hình 23: API lấy danh sách lớp học

```
@app.route('/api/classes', methods=['POST'])
def create_class():
    """Tạo lớp học mới"""
    try:
        # Kiểm tra quyền
        if not session.get('logged_in'):
            return jsonify({
                'success': False,
                'message': 'Vui lòng đăng nhập'
            }), 401
        if not check_permission('classes.create'):
            return jsonify({
                'success': False,
                'message': 'Bạn không có quyền thêm lớp học mới'
            }), 403
        data = request.get_json()
        required_fields = ['class_code', 'class_name']
        for field in required_fields:
            if not data.get(field):
                return jsonify({
                    'success': False,
                    'message': f'Trường {field} là bắt buộc'
                }), 400
        connection = connect_database()
        if not connection:
            return jsonify({
                'success': False,
                'message': 'Lỗi kết nối database'
            }), 500
        try:
            cursor = connection.cursor(dictionary=True)
            # Kiểm tra mã lớp đã tồn tại
            check_query = "SELECT id FROM attendance_classes WHERE class_code = %s"
            cursor.execute(check_query, (data['class_code'],))
            if cursor.fetchone():
                return jsonify({
                    'success': False,
                    'message': 'Mã lớp đã tồn tại'
                }), 400
            # Thêm lớp học mới
            current_time = int(datetime.now().timestamp())

            insert_query = """
            INSERT INTO attendance_classes
            (class_code, class_name, status, created_time, updated_time)
            VALUES (%s, %s, %s, %s, %s)
            """
            values = [
                data['class_code'],
                data['class_name'],
                data.get('status', 1),
                current_time,
                current_time
            ]
            cursor.execute(insert_query, values)
            connection.commit()

            class_id = cursor.lastrowid

            return jsonify({
                'success': True,
                'message': 'Thêm lớp học thành công',
                'data': {
                    'id': class_id,
                    'class_code': data['class_code']
                }
            })

        finally:
            cursor.close()
            connection.close()

    except Exception as e:
        return jsonify({
            'success': False,
            'message': 'Lỗi server nội bộ'
        }), 500
```

Hình 24: API tạo mới lớp học

### 3.2.2.2. API quản lý sinh viên

```
@app.route('/api/students', methods=['POST'])
def create_student():
    """Tạo sinh viên mới"""
    try:
        # Kiểm tra quyền
        if not session.get('logged_in'):
            return jsonify({
                'success': False,
                'message': 'Vui lòng đăng nhập'
            }), 401

        if not check_permission('students.create'):
            return jsonify({
                'success': False,
                'message': 'Bạn không có quyền thêm sinh viên mới'
            }), 403

        data = request.get_json()

        required_fields = ['student_code', 'full_name']
        for field in required_fields:
            if not data.get(field):
                return jsonify({
                    'success': False,
                    'message': f'Trường {field} là bắt buộc'
                }), 400

        connection = connect_database()
        if not connection:
            return jsonify({
                'success': False,
                'message': 'Lỗi kết nối database'
            }), 500

        try:
            cursor = connection.cursor(dictionary=True)

            # Kiểm tra mã sinh viên đã tồn tại
            check_query = "SELECT id FROM students WHERE student_code = %s"
            cursor.execute(check_query, (data['student_code'],))
            if cursor.fetchone():
                return jsonify({
                    'success': False,
                    'message': 'Mã sinh viên đã tồn tại'
                }), 400

            # Thêm sinh viên mới
            current_time = int(datetime.now().timestamp())

            insert_query = """
            INSERT INTO students
            (student_code, full_name, class_name, status, created_time, updated_time)
            VALUES (%s, %s, %s, %s, %s, %s)
            """

            values = {
                'student_code': data['student_code'],
                'full_name': data['full_name'],
                'class_name': data.get('class_name', ''),
                'status': data.get('status', 1),
                'created_time': current_time,
                'updated_time': current_time
            }

            cursor.execute(insert_query, values)
            connection.commit()

            student_id = cursor.lastrowid

            # Xử lý ảnh khuôn mặt nếu có
            if data.get('images_base64'):
                save_face_images(data['student_code'], data['images_base64'])

            return jsonify({
                'success': True,
                'message': 'Thêm sinh viên thành công',
                'data': {
                    'id': student_id,
                    'student_code': data['student_code']
                }
            })

        finally:
            cursor.close()
            connection.close()

    except Exception as e:
        return jsonify({
            'success': False,
            'message': 'Lỗi server nội bộ'
        }), 500
```

Hình 25: API tạo mới sinh viên

```
@app.route('/api/students', methods=['GET'])
def get_students():
    """Lấy danh sách sinh viên"""
    try:
        connection = connect_database()
        if not connection:
            return jsonify({
                'success': False,
                'message': 'Lỗi kết nối database'
            }), 500

        try:
            cursor = connection.cursor(dictionary=True)

            # Lấy tất cả sinh viên cho client-side pagination
            search = request.args.get('search', '')

            where_clause = "WHERE status = 1"
            params = []

            if search:
                where_clause += " AND (student_code LIKE %s OR full_name LIKE %s OR room_number LIKE %s)"
                search_param = f"%{search}%"
                params.extend([search_param, search_param, search_param])

            # Đếm tổng số bản ghi
            count_query = f"SELECT COUNT(*) as total FROM students {where_clause}"
            cursor.execute(count_query, params)
            total = cursor.fetchone()['total']

            # Lấy tất cả dữ liệu
            query = f"""
                SELECT id, student_code, full_name, class_name, created_time, is_trained
                FROM students
                {where_clause}
                ORDER BY created_time DESC
            """
            cursor.execute(query, params)
            students = cursor.fetchall()

            # Format dữ liệu
            for student in students:
                if student.get('created_time'):
                    student['created_time'] = datetime.fromtimestamp(student['created_time']).strftime('%Y-%m-%d %H:%M:%S')

            return jsonify({
                'success': True,
                'data': {
                    'students': students,
                    'total': total
                }
            })

        finally:
            cursor.close()
            connection.close()

    except Exception as e:
        return jsonify({
            'success': False,
            'message': 'Lỗi server nội bộ'
        }), 500
```

Hình 26: API lấy danh sách sinh viên

### 3.2.2.3. API nhận diện

```
# Xử lý nhận diện
results = process_frame(frame)
# Phát hiện khuôn mặt trong frame
bboxes, landmarks = detector.detect(frame)
faces_detected = len(bboxes) if bboxes is not None else 0
# Kiểm tra kết quả phát hiện khuôn mặt
if results:
    auto_log_access = False
    access_type = 1 # Mặc định: Vào
    if request.content_type and 'application/json' in request.content_type:
        data = request.json
        auto_log_access = data.get('auto_log_access', False)
        access_type = data.get('access_type', 1)
    if auto_log_access and results:
        for result in results:
            if result.get('student_code') and result.get('score', 0) >= FACE_RECOGNITION_CONFIG['auto_log_threshold']: # Chỉ log nếu score cao
                try:
                    # Ghi nhận ra vào tự động với logic thông minh
                    connection = connect_database()
                    if connection:
                        try:
                            cursor = connection.cursor(dictionary=True)
                            # Kiểm tra sinh viên tồn tại
                            student_query = "SELECT id FROM students WHERE student_code = %s AND status = 1"
                            cursor.execute(student_query, (result['student_code'],))
                            student = cursor.fetchone()
                            if student:
                                current_time = int(datetime.now().timestamp())
                                # **LOGIC THÔNG MINH: Kiểm tra trạng thái vào/ra gần nhất**
                                last_access_query = """
                                SELECT access_type, access_time
                                FROM attendance_access_logs
                                WHERE student_code = %s
                                ORDER BY access_time DESC
                                LIMIT 1
                                """
                                cursor.execute(last_access_query, (result['student_code'],))
                                last_access = cursor.fetchone()

                                # Xác định access_type dựa trên lần cuối
                                if last_access:
                                    # Nếu lần cuối là "Vào" (1) → Lần này là "Ra" (2)
                                    # Nếu lần cuối là "Ra" (2) → Lần này là "Vào" (1)
                                    smart_access_type = 2 if last_access['access_type'] == 1 else 1
                                    action_text = "ra khỏi" if smart_access_type == 2 else "vào"
                                else:
                                    # Lần đầu tiên → Mặc định là "Vào"
                                    smart_access_type = 1
                                    action_text = "vào"
                                # Ghi log với access_type thông minh
                                insert_query = """
                                INSERT INTO attendance_access_logs
                                (student_id, student_code, access_type, access_time, gate_location,
                                verification_method, device_id, device_name, ip_address,
                                recognition_rate, notes, status, created_time)
                                VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)
                                """
                                values = (
                                    student['id'],
                                    result['student_code'],
                                    smart_access_type, # Sử dụng access_type thông minh
                                    current_time,
                                    data.get('gate_location', 'Cổng chính'),
                                    'Nhận diện khuôn mặt',
                                    data.get('device_id', 'FACE_RECOGNITION_CAM'),
                                    data.get('device_name', 'Camera nhận diện khuôn mặt'),
                                    request.remote_addr,
                                    result['score'] * 100, # Convert to percentage
                                    f'Nhận diện tự động - {action_text} KTX - Score: {result['score']:.2f}',
                                    1,
                                    current_time
                                )
                                cursor.execute(insert_query, values)
                                connection.commit()
```

Hình 27: API nhận diện khuôn mặt

```
@app.route('/api/access-logs', methods=['GET'])
def get_access_logs():
    """Lấy danh sách lịch sử ra/vào"""
    try:
        connection = connect_database()
        if not connection:
            return jsonify({
                'success': False,
                'message': 'Lỗi kết nối database'
            }), 500
        try:
            cursor = connection.cursor(dictionary=True)
            page = int(request.args.get('page', 1))
            limit = int(request.args.get('limit', 50))
            offset = (page - 1) * limit
            query = """
                SELECT al.id, al.student_code,
                       FROM_UNIXTIME(al.access_time) as access_time,
                       al.access_type, al.gate_location,
                       al.verification_method, al.recognition_rate, al.notes,
                       s.full_name as student_name, s.room_number
                FROM attendance_access_logs al
                LEFT JOIN students s ON al.student_code = s.student_code
                ORDER BY al.access_time DESC
                LIMIT %s OFFSET %s
            """
            cursor.execute(query, (limit, offset))
            logs = cursor.fetchall()
            for log in logs:
                if log['access_time'] and hasattr(log['access_time'], 'strftime'):
                    log['access_time'] = log['access_time'].strftime('%Y-%m-%d %H:%M:%S')
                if log['access_type'] == 1:
                    log['access_type'] = 'entry'
                elif log['access_type'] == 2:
                    log['access_type'] = 'exit'
            return jsonify({
                'success': True,
                'data': logs
            })
        finally:
            cursor.close()
            connection.close()
    except Exception as e:
        error_msg = str(e)
        if "doesn't exist" in error_msg or "Table" in error_msg:
            return jsonify({
                'success': True,
                'data': [],
                'message': 'Bảng access logs chưa được tạo. Hệ thống sẽ tạo tự động khi có dữ liệu đầu tiên.'
            })
        return jsonify({
            'success': False,
            'message': f'Lỗi server nội bộ: {error_msg}'
        }), 500
```

Hình 28: API lấy danh sách điểm danh

### 3.3. Xây dựng frontend - giao diện người dùng

Phần frontend được xây dựng nhằm cung cấp giao diện trực quan, dễ sử dụng cho người dùng hệ thống như: quản trị viên, giảng viên và sinh viên. Các mục tiêu chính bao gồm:

- Hiển thị đầy đủ và chính xác các thông tin quản lý lớp học, sinh viên và buổi điểm danh.
- Cung cấp giao diện để thao tác với các chức năng CRUD (thêm, sửa, xóa).
- Tích hợp giao diện điểm danh tự động theo thời gian thực.
- Tối ưu trải nghiệm người dùng (UX) với bố cục rõ ràng, tốc độ xử lý nhanh.
- Đảm bảo giao diện có thể hoạt động tốt trên nhiều độ phân giải khác nhau.

```

<body>
  <!-- Mobile Menu Toggle -->
  <button class="mobile-menu-toggle" id="mobile-menu-toggle">
    <i class="fas fa-bars"></i>
  </button>

  <div class="container">
    <!-- Menu Container - loaded from menu.html -->
    <div id="menu-container"></div>

    <!-- Main Content -->
    <div class="main-content">
      <!-- Header -->
      <div class="header">
        <div class="header-left">
          <h1>Điểm danh sinh viên</h1>
          <div class="date-time">
            <span id="current-date">Thứ Hai, 20 tháng 10, 2025</span>
            <span id="current-time">13:00:00</span>
          </div>
        </div>
      </div>

      <!-- Checkin Section Container -->
      <div id="checkin-section-container"></div>
    </div>
  </div>

  <script>
    // Global variables
    if (typeof API_BASE_URL === 'undefined') {
      const API_BASE_URL = window.location.origin;
    }

    // Update date and time
    function updateDateTime() {
      const now = new Date();
      const options = { weekday: 'long', year: 'numeric', month: 'long', day: 'numeric' };
      const dateStr = now.toLocaleDateString('vi-VN', options);
      const timeStr = now.toLocaleTimeString('vi-VN');

      const dateEl = document.getElementById('current-date');
      const timeEl = document.getElementById('current-time');

      if (dateEl) dateEl.textContent = dateStr;
      if (timeEl) timeEl.textContent = timeStr;
    }

    // Load menu from menu.html
    function loadMenu() {
      fetch('/static/menu.html')
        .then(response => response.text())
        .then(html => {
          document.getElementById('menu-container').innerHTML = html;
          // Execute scripts in loaded HTML
          const scripts = document.getElementById('menu-container').querySelectorAll('script');
          scripts.forEach(script => {
            const newScript = document.createElement('script');
            newScript.textContent = script.textContent;
            document.body.appendChild(newScript);
          });
        })
        .catch(error => console.error('Error loading menu:', error));
    }

    // Load checkin section
    function loadCheckinSection() {
      fetch('/static/checkin.html')
        .then(response => response.text())
        .then(html => {
          document.getElementById('checkin-section-container').innerHTML = html;
          // Add active class to show the section
          const section = document.querySelector('#checkin-section-container .checkin-section');
          if (section) {
            section.classList.add('active');
          }
          // Execute scripts in loaded HTML
          const scripts = document.getElementById('checkin-section-container').querySelectorAll('script');
          scripts.forEach(script => {
            const newScript = document.createElement('script');
            newScript.textContent = script.textContent;
            document.body.appendChild(newScript);
          });
        })
        .catch(error => console.error('Error loading checkin section:', error));
    }

    // Load section when DOM is ready
    document.addEventListener('DOMContentLoaded', function() {
      updateDateTime();
      setInterval(updateDateTime, 1000);
      loadMenu();
      loadCheckinSection();
    });
  </script>

```

Hình 29: Mã code Giao diện điểm danh

### 3.4. Tích hợp mô hình nhận diện khuôn mặt vào hệ thống

Mục tiêu của phần này là tích hợp mô hình nhận diện khuôn mặt vào hệ thống điểm danh nhằm:

- Tự động phát hiện và nhận diện sinh viên thông qua hình ảnh từ camera.
- Chuyển đổi khuôn mặt thành vector đặc trưng (embedding) để so khớp với dữ liệu đã lưu.
- Ghi nhận điểm danh vào/ra theo thời gian thực.
- Tối ưu độ chính xác và hiệu năng xử lý.
- Việc tích hợp mô hình giúp hệ thống hoạt động tự động, giảm tối đa sự can thiệp thủ công.

Các mô hình được sử dụng: Mô hình phát hiện khuôn mặt – SCRFD, Mô hình nhận diện – ArcFace.

Các bước xử lý nhận diện khuôn mặt:

Bước 1: Nhận ảnh từ frontend

Bước 2: Phát hiện khuôn mặt, mô hình SCRFD xác định khuôn mặt trong ảnh

```
# Face detector
detector = SCRFD(model_file="face_detection/scrfd/weights/scrfd_2.5g_bnkp.onnx")
```

Hình 30: Đoạn mã khởi tạo mô hình phát hiện khuôn mặt SCRFD.

Bước 3: Chuẩn hóa khuôn mặt

```
# Căn chỉnh khuôn mặt
face_aligned = norm_crop(frame, landmark)

# Chuyển đổi từ BGR sang RGB
face_aligned = cv2.cvtColor(face_aligned, cv2.COLOR_BGR2RGB)

# Chuyển đổi từ HWC sang CHW
face_aligned = face_aligned.transpose(2, 0, 1)

# Trích xuất đặc trưng
face_tensor = torch.from_numpy(face_aligned).float()
face_tensor = face_tensor.to(device)
face_tensor = face_tensor.unsqueeze(0) # Thêm batch dimension

# Chuẩn hóa ảnh
face_tensor = (face_tensor - 127.5) / 128.0
```

Hình 31: Đoạn mã xử lý và chuẩn hóa ảnh khuôn mặt trước khi nhận diện.

Bước 4: Sinh embedding từ ArcFace

```
# Face recognizer
recognizer = iresnet_inference(
    model_name="r100",
    path="face_recognition/arcface/weights/arcface_r100.pth",
    device=device
)
```

Hình 32: Đoạn mã xử lý và chuẩn hóa ảnh khuôn mặt trước khi nhận diện.

Bước 5: So khớp với database

```
# So sánh với database
if features is not None:
    score, idx = compare_encodings(embedding, images_embs)
    folder_name = images_names[idx]
    score = float(score[0])
```

Hình 33: Đoạn mã so sánh đặc trưng khuôn mặt với cơ sở dữ liệu.

Bước 6: Trả kết quả và ghi log điểm danh

```
# Tạo thông báo dựa trên kết quả
message = ""
if faces_detected == 0:
    message = "Không phát hiện khuôn mặt nào trong frame"
elif len(results) == 0 and faces_detected > 0:
    message = f"Phát hiện {faces_detected} khuôn mặt nhưng không nhận diện được (score thấp hoặc không có trong database)"
elif len(results) > 0:
    message = f"Nhận diện thành công {len(results)} khuôn mặt"
```

Hình 34: Đoạn mã trả về kết quả.

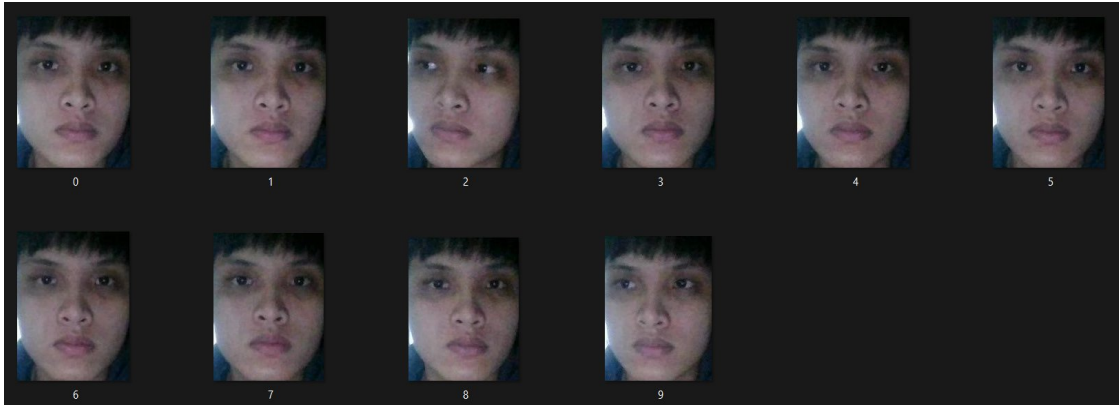
### 3.5. Triển khai hệ thống trên môi trường thực nghiệm

Mục tiêu của bước triển khai là đưa toàn bộ hệ thống (Frontend – Backend – AI – Database) vào hoạt động thực tế để kiểm thử chức năng, hiệu năng và khả năng đáp ứng trước khi áp dụng vào môi trường sản xuất.

- Kiểm tra tính ổn định của API backend khi xử lý dữ liệu thời gian thực.
- Kiểm thử khả năng nhận diện khuôn mặt trong điều kiện ánh sáng và môi trường khác nhau.
- Đánh giá tốc độ phản hồi khi điểm danh tự động.
- Xác thực tính chính xác và độ tin cậy của luồng dữ liệu từ Camera → Backend → Database.
- Đảm bảo hệ thống vận hành liên tục trong thời gian dài.

Kết quả khi triển khai thực nghiệm:

- Độ chính xác (Accuracy): Tỷ lệ nhận diện đúng 92% – 98% (tùy ánh sáng & góc mặt), nhận sai thấp: < 3% khi đặt threshold 0.75–0.85.
- Tốc độ xử lý: Đáp ứng tốt yêu cầu real-time.
- Hiệu năng ổn định: Chạy liên tục không rò rỉ bộ nhớ, API giữ tốc độ ổn định dưới tải 20–30 requests/giây.
- Trải nghiệm người dùng: Frontend hiển thị kết quả nhận diện gần như tức thì, thời gian điểm danh mỗi sinh viên < 1 giây.



*Hình 35: Tập dữ liệu khuôn mặt thu thập cho quá trình huấn luyện mô hình.*

Các vấn đề gặp phải:

- Ánh sáng yếu, mô hình khó nhận diện → tăng độ sáng tại khu vực camera.
- Góc mặt khó
- Nhiều môi trường (nhiều người, chuyển động)
- Camera chất lượng thấp

Triển khai hệ thống trên môi trường thực nghiệm:

- Hoạt động ổn định và đáp ứng tốt yêu cầu thời gian thực.
- Nhận diện chính xác cao khi điều kiện ánh sáng phù hợp.
- Ghi log điểm danh tự động, chính xác, giảm tối đa thao tác thủ công.
- Sẵn sàng để triển khai trong môi trường thực tế có số lượng sinh viên lớn.

Quá trình triển khai thực nghiệm đã chứng minh tính khả thi và hiệu quả thực tế của hệ thống điểm danh nhận diện khuôn mặt. Với độ chính xác đạt từ 92% đến 98% và thời gian phản hồi dưới 1 giây cho mỗi lượt điểm danh, hệ thống đã đáp ứng tốt các yêu cầu cốt lõi về tốc độ và độ tin cậy trong môi trường thời gian thực.

Mặc dù vẫn tồn tại một số thách thức khách quan liên quan đến điều kiện môi trường (ánh sáng yếu, góc chụp khó), nhưng các thử nghiệm cũng chỉ ra rằng việc chuẩn hóa vị trí lắp đặt và điều chỉnh thông số kỹ thuật (threshold) có thể khắc phục đáng kể các hạn chế này. Nhìn chung, sự ổn định của API backend dưới áp lực tải và khả năng vận

hành liên tục khẳng định hệ thống đã hoàn thiện về mặt kỹ thuật, sẵn sàng chuyển từ giai đoạn thực nghiệm sang triển khai ứng dụng thực tế quy mô lớn.

## KẾT LUẬN

### 1. Kết luận chung

- Đề tài đã nghiên cứu và ứng dụng thành công các kỹ thuật thị giác máy tính và trí tuệ nhân tạo vào bài toán điểm danh sinh viên.
- Hệ thống web được triển khai hoàn chỉnh, bao gồm giao diện người dùng, backend xử lý và cơ sở dữ liệu quản lý thông tin.
- Mô hình nhận diện khuôn mặt hoạt động ổn định, đáp ứng tốt yêu cầu điểm danh theo thời gian thực trong môi trường lớp học.

### 2. Kết quả đạt được

- Xây dựng hệ thống nhận diện khuôn mặt dựa trên SCRFD (phát hiện) và ArcFace + iResNet-100 (trích xuất đặc trưng), đạt độ chính xác cao.
- Hoàn thiện các chức năng chính: quản lý sinh viên, quản lý lớp học, điểm danh bằng camera, lưu trữ lịch sử điểm danh và xem báo cáo.
- Kết hợp thành công Flask, MySQL và giao diện web HTML/CSS/JavaScript để tạo thành hệ thống hoàn chỉnh, dễ sử dụng và dễ mở rộng.
- Tối ưu quy trình điểm danh giúp giảm thời gian và hạn chế gian lận so với điểm danh thủ công.

### 3. Đóng góp của đề tài

- Mang lại giải pháp tự động hóa quy trình điểm danh bằng công nghệ AI, phù hợp với xu hướng chuyển đổi số trong giáo dục.
- Góp phần cải thiện tính chính xác và minh bạch trong quản lý chuyên cần của sinh viên.
- Cung cấp nền tảng có thể mở rộng cho các hệ thống quản lý học tập khác như kiểm soát ra vào, thống kê buổi học, hoặc báo cáo chuyên cần tự động.

### 4. Đề xuất và kiến nghị

- Trong tương lai có thể tích hợp thêm mô hình chống giả mạo (anti-spoofing) để tránh sử dụng ảnh hoặc video để đánh lừa hệ thống.
- Tối ưu hiệu suất để hệ thống chạy mượt trên các thiết bị cấu hình thấp.
- Triển khai giao diện người dùng hiện đại hơn và tích hợp ứng dụng di động để thuận tiện cho việc điểm danh từ nhiều thiết bị.

## **TÀI LIỆU THAM KHẢO**

### **Tiếng Việt**

- [1] Trần Minh Triết, Nguyễn Thanh Bình, Giáo trình Trí tuệ nhân tạo, NXB Đại học Quốc gia TP. Hồ Chí Minh, 2019.
- [2] P. N. Minh, “Ứng dụng thị giác máy tính trong nhận dạng đối tượng,” Tạp chí CNTT & Truyền thông, số 4, tr. 30–38, 2021.
- [3] Nguyễn Đức Thuận, Ứng dụng học máy trong thị giác máy tính, Tạp chí Công nghệ Thông tin và Truyền thông, số 3, tr. 45-52, 2021.
- [4] Nguyễn Văn Vinh, Xử lý ảnh số và ứng dụng trong nhận dạng, NXB Khoa học và Kỹ thuật, 2020.

### **Tiếng Anh**

- [5] G. Bradski and A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library. Sebastopol, CA, USA: O’Reilly Media, 2008.
- [6] J. Guo, J. Deng, A. Lattas, and S. Zafeiriou, “SCRFD: Towards efficient face detection via sparse regression and CNN optimization,” arXiv preprint arXiv:2105.04714, 2021.
- [7] M. Grinberg, Flask Web Development: Developing Web Applications with Python, 2nd ed. Sebastopol, CA, USA: O’Reilly Media, 2018.

