|  |
| --- |
| **BỘ GIAO THÔNG VẬN TẢI**  **HỌC VIỆN HÀNG KHÔNG VIỆT NAM**  **KHOA CÔNG NGHỆ THÔNG TIN**      **ĐỒ ÁN CHUYÊN NGÀNH**  **CÔNG NGHỆ THÔNG TIN**  **XÂY DỰNG HỆ THỐNG ĐIỂM DANH SINH VIÊN BẰNG KHUÔN MẶT**  **Giảng viên hướng dẫn:Th.S Trần Anh Tuấn**  **Sinh viên/ Nhóm sinh viên thực hiện: Nhóm 06**  **Mã số sinh viên:**  **Lớp: 010100086405**  **Thành phố Hồ Chí Minh, tháng 12/2024** |
| **BỘ GIAO THÔNG VẬN TẢI**  **HỌC VIỆN HÀNG KHÔNG VIỆT NAM**  **KHOA CÔNG NGHỆ THÔNG TIN**      **ĐỒ ÁN CHUYÊN NGÀNH**  **CÔNG NGHỆ THÔNG TIN**  **XÂY DỰNG HỆ THỐNG ĐIỂM DANH SINH VIÊN BẰNG KHUÔN MẶT**  **Giảng viên hướng dẫn:Th.S Trần Anh Tuấn**  **Sinh viên/ Nhóm sinh viên thực hiện: Nhóm 06**  **Mã số sinh viên:**  **Lớp: 010100086405**  **Thành phố Hồ Chí Minh, tháng 12/2024** |

**Danh sách Nhóm:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Lớp** | **Ghi chú** |
| 1 | Trần Khang | 2254810213 | 22ĐHTT05 | Nhóm Trưởng |
| 2 | Võ Văn Trình | 2254810220 | 22ĐHTT05 | Thành Viên |
| 3 | Trần Thanh Đạt | 2254810182 | 22ĐHTT04 | Thành Viên |
| 4 | Nguyễn Khánh Duy | 2254810157 | 22ĐHTT04 | Thành Viên |
| 5 | Nguyễn Phạm Thùy Dung | 2254810131 | 22ĐHTT04 | Thành Viên |

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

# **DANH MỤC HÌNH ẢNH**

[Hình 3.1. Usecase tổng quát 23](#_Toc184044304)

[Hình 3.2. Usecase sinh viên 23](#_Toc184044305)

[Hình 3.3. Usecase tính điểm chuyên cần 24](#_Toc184044306)

[Hình 3.4. Usecase lớp học phần 24](#_Toc184044307)

[Hình 3.5. Usecase lịch sử điểm danh 25](#_Toc184044308)

[Hình 3.6. Usecase lớp học phần 25](#_Toc184044309)

[Hình 3.7. Sơ đồ tuần tự xóa sinh viên 26](#_Toc184044310)

[Hình 3.8. Sơ đồ tuần tự thêm sinh viên 27](#_Toc184044311)

[Hình 3.9. Sơ đồ hoạt động của cập nhật sinh viên 28](#_Toc184044312)

[Hình 3.10. Sơ đồ hoạt động của thêm sinh viên 29](#_Toc184044313)

[Hình 3.11. Sơ đồ hoạt động của xóa sinh viên 30](#_Toc184044314)

[Hình 3.12. Database 31](#_Toc184044315)

[Hình 3.13. Giao diện của tính năng sinh viên 33](#_Toc184044316)

[Hình 3.14. Giao diện của tính năng điểm danh 34](#_Toc184044317)

[Hình 3.15. Giao diện của tính năng lịch sử điểm danh 35](#_Toc184044318)

[Hình 3.16. Giao diện của tính năng chức năng lớp học phần 36](#_Toc184044319)

[Hình 3.17. Giao diện của tính năng tính điểm chuyên cần 37](#_Toc184044320)

**MỤC LỤC**

[**DANH MỤC HÌNH ẢNH** 4](#_Toc184045973)

[**MỞ ĐẦU** 7](#_Toc184045974)

[**CHƯƠNG 1. GIỚI THIỆU** 8](#_Toc184045975)

[**1.1. Lý do chọn đề tài** 8](#_Toc184045976)

[**1.2. Mục tiêu đề tài** 8](#_Toc184045977)

[**1.3. Phạm vi đề tài** 8](#_Toc184045978)

[**1.4 Đối tượng nghiên cứu** 9](#_Toc184045979)

[**1.5. Phương pháp nghiên cứu** 10](#_Toc184045980)

[**1.5.1. Phương pháp thu thập thông tin** 10](#_Toc184045981)

[**1.5.2. Phương pháp xử lý thông tin** 10](#_Toc184045982)

[**1.5.3. Phương pháp thực nghiệm** 10](#_Toc184045983)

[**1.5.4. Phương pháp thiết kế và mô phỏng** 11](#_Toc184045984)

[**1.5.5. Phương pháp thống kê** 11](#_Toc184045985)

[**1.5.6. Phương pháp so sánh và đánh giá** 11](#_Toc184045986)

[**1.5.7. Phương pháp chuyên gia** 11](#_Toc184045987)

[**1.6. Bố cục đề tài** 11](#_Toc184045988)

[**CHƯƠNG 2. CƠ SỞ LÝ THUYẾT** 12](#_Toc184045989)

[**2.1. PYTHON** 12](#_Toc184045990)

[**2.1.1. Đơn giản, dễ học và dễ đọc** 12](#_Toc184045991)

[**2.1.2. Đa năng** 12](#_Toc184045992)

[**2.1.3. Mã nguồn mở** 13](#_Toc184045993)

[**2.1.4. Thư viện phong phú** 13](#_Toc184045994)

[**2.1.5. Di động** 13](#_Toc184045995)

[**2.1.6. Hướng đối tượng** 13](#_Toc184045996)

[**2.1.7. Tính tương tác** 13](#_Toc184045997)

[**2.1.8. Tính bảo mật và khả năng mở rộng** 13](#_Toc184045998)

[**2.1.8. Ưu điểm của Python** 13](#_Toc184045999)

[**2.1.9. Nhược điểm của Python** 14](#_Toc184046000)

[**2.2. XAMP** 15](#_Toc184046001)

[**2.2.1. Xampp là gì và công dụng** 15](#_Toc184046002)

[**2.2.2. Chức năng của xampp** 15](#_Toc184046003)

[**2.2.3. Ưu điểm và nhược điểm của xampp** 15](#_Toc184046004)

[**2.3. QT DESIGNER** 16](#_Toc184046005)

[**2.3.1. Qt Designer là gì** 16](#_Toc184046006)

[**2.3.2. Chức năng của Qt Designer** 16](#_Toc184046007)

[**2.3.3. Công dụng của Qt Designer** 16](#_Toc184046008)

[**2.3.4. Ưu điểm và nhược điểm của Qt Designer** 17](#_Toc184046009)

[**2.4. Thuật toán Haar-cascade** 17](#_Toc184046010)

[**2.4.1. Thuật toán Haar Cascade là gì?** 17](#_Toc184046011)

[**2.4.2. Cấu trúc và nguyên lý hoạt động của thuật toán Haar Cascade** 17](#_Toc184046012)

[**2.4.2.1. Haar Features (đặc trưng Haar)** 18](#_Toc184046013)

[**2.4.2.2. Integral Image (hình ảnh tổng hợp)** 18](#_Toc184046014)

[**2.4.2.3. Cascade Classifier (bộ phân loại Cascade)** 18](#_Toc184046015)

[**2.4.2.4. Huấn luyện bộ phân loại** 18](#_Toc184046016)

[**2.4.2.5. Quá trình quét (Sliding Window)** 18](#_Toc184046017)

[**2.4.3. Ưu điểm và nhược điểm của Haar Cascade** 18](#_Toc184046018)

[**2.5. Mô hình Nhận diện LBPH Recognizer** 19](#_Toc184046019)

[**2.5.1. Tổng quan về LBPH (Local Binary Patterns Histogram)** 19](#_Toc184046020)

[**2.5.2. Quá trình nhận diện với LBPH Recognizer** 20](#_Toc184046021)

[**2.6. Thuật toán LBPH (Local Binary Patterns Histograms)** 20](#_Toc184046022)

[**CHƯƠNG 3. PHÂN TÍCH HỆ THỐNG VÀ XÂY DỰNG SẢN PHẨM** 22](#_Toc184046023)

[**3.1. Phân tích hệ thống** 22](#_Toc184046024)

[**3.1.1 Sơ Đồ Usecase** 22](#_Toc184046025)

[**3.1.2. Sơ Đồ Tuần Tự** 26](#_Toc184046026)

[**3.1.3. Sơ Đồ Hoạt Động** 28](#_Toc184046027)

[**3.2. Xây dựng giao diện sản phẩm** 31](#_Toc184046028)

[**3.2.1 Chức Năng Sản Phẩm** 31](#_Toc184046029)

[**3.2.3 Giao Diện Sản Phẩm** 33](#_Toc184046030)

[**KẾT LUẬN** 38](#_Toc184046031)

[**DANH MỤC TÀI LIỆU THAM KHẢO** 39](#_Toc184046032)

# **MỞ ĐẦU**

Trong bối cảnh công nghệ thông tin ngày càng phát triển, việc ứng dụng các công nghệ tiên tiến vào quản lý và giảng dạy trở nên ngày càng phổ biến và cần thiết. Công nghiệp 4.0 đã tác động mạnh mẽ đến mọi lĩnh vực của cuộc sống, đặc biệt là trong lĩnh vực giáo dục. Việc ứng dụng các giải pháp công nghệ vào quản lý và giảng dạy không chỉ giúp nâng cao chất lượng giáo dục mà còn tối ưu hóa các quy trình quản lý, tiết kiệm thời gian và nguồn lực. Nhận diện khuôn mặt, một trong những công nghệ tiên tiến nhất hiện nay, đã và đang được ứng dụng rộng rãi trong nhiều lĩnh vực, bao gồm an ninh, chăm sóc sức khỏe và giáo dục.Xuất phát từ nhu cầu thực tế về việc tối ưu hóa quy trình điểm danh trong các lớp học, nhóm chúng em đã quyết định chọn đề tài "Xây dựng hệ thống điểm danh và dự đoán thời gian đi học của sinh viên bằng khuôn mặt sử dụng camera điện thoại". Đề tài này nhằm mục tiêu ứng dụng công nghệ nhận diện khuôn mặt để tự động hóa quy trình điểm danh, từ đó giúp các giảng viên dễ dàng quản lý sĩ số lớp học và theo dõi tình trạng sinh viên một cách hiệu quả hơn. Đồng thời, hệ thống cũng cung cấp tính năng dự đoán thời gian sinh viên đến lớp dựa trên dữ liệu lịch sử điểm danh, giúp giảng viên có cái nhìn tổng quan về xu hướng đi học của sinh viên.Trong quá trình thực hiện đề tài, chúng em đã nỗ lực nghiên cứu, phân tích và triển khai các phương pháp xử lý hình ảnh, nhận diện khuôn mặt và dự đoán dữ liệu. Mặc dù đã cố gắng hoàn thiện, nhưng do hạn chế về thời gian và nguồn lực, chắc chắn đồ án không tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự góp ý, chỉ dẫn từ quý thầy cô và các bạn để đề tài được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn sự hướng dẫn tận tình của giảng viên hướng dẫn, cùng với sự hỗ trợ từ các bạn bè đã góp phần giúp chúng em hoàn thành đồ án này.

# **CHƯƠNG 1. GIỚI THIỆU**

## **1.1. Lý do chọn đề tài**

Hiện nay, các trường học, đặc biệt là các trường đại học, đang phải đối mặt với vấn đề quản lý điểm danh sinh viên một cách hiệu quả. Phương pháp điểm danh truyền thống, như gọi tên hay quét thẻ, thường tốn nhiều thời gian, dễ gây nhầm lẫn hoặc gian lận, ví dụ như sinh viên có thể nhờ bạn điểm danh hộ. Thực tế này đòi hỏi một giải pháp tự động và chính xác hơn để đảm bảo việc quản lý thời gian học và sự hiện diện của sinh viên.Việc xây dựng một hệ thống điểm danh tự động bằng nhận diện khuôn mặt sẽ giải quyết các hạn chế của phương pháp điểm danh thủ công, cung cấp một cách quản lý nhanh chóng và chính xác hơn. Sự phát triển của công nghệ trí tuệ nhân tạo (AI) và học sâu (Deep Learning) cho phép nhận diện khuôn mặt trở nên chính xác và có tính ứng dụng cao hơn, đặc biệt là trong lĩnh vực giáo dục.Nếu vấn đề này được giải quyết, hệ thống sẽ giúp tiết kiệm thời gian, công sức của cả giáo viên và sinh viên, đảm bảo tính chính xác và minh bạch trong việc điểm danh. Ngoài ra, hệ thống còn góp phần vào việc theo dõi và quản lý thời gian học tập của sinh viên một cách hiệu quả, từ đó nâng cao chất lượng giáo dục.Hệ thống này không chỉ giới hạn trong lĩnh vực giáo dục mà còn có thể được áp dụng vào nhiều lĩnh vực khác như quản lý nhân sự tại các công ty, quản lý sự kiện, hội nghị hay trong các hệ thống an ninh tại các tòa nhà, công ty, khu vực công cộng.

## **1.2. Mục tiêu đề tài**

Mục tiêu của đề tài là phát triển một hệ thống sử dụng công nghệ nhận diện khuôn mặt để tự động điểm danh sinh viên. Hệ thống này được thiết kế dành cho giảng viên và quản lý lớp học, giúp tiết kiệm thời gian, giảm thiểu sai sót trong quá trình điểm danh, đồng thời đảm bảo tính minh bạch và công bằng cho sinh viên. Hệ thống cần đáp ứng khả năng quét và xác minh danh tính sinh viên dựa trên dữ liệu khuôn mặt đã lưu trữ, ghi nhận thông tin điểm danh và cung cấp báo cáo về mức độ tham gia của sinh viên. Về mặt công nghệ, hệ thống sẽ ứng dụng ngôn ngữ lập trình Python và các thư viện xử lý hình ảnh như OpenCV, kết hợp với các thuật toán học máy hoặc học sâu để đảm bảo độ chính xác cao trong việc nhận diện. Ngoài ra, hệ thống sẽ được triển khai trên nền tảng thân thiện với người dùng, có khả năng quản lý dữ liệu hiệu quả và cung cấp giao diện trực quan để hỗ trợ việc thêm, xóa và cập nhật dữ liệu sinh viên.

## **1.3. Phạm vi đề tài**

Đề tài sẽ được thực hiện để phát triển một hệ thống ứng dụng công nghệ nhận diện khuôn mặt để tự động điểm danh sinh viên trong môi trường giáo dục. Hệ thống sẽ được triển khai tại các cơ sở giáo dục như trường đại học hoặc cao đẳng, tập trung vào việc áp dụng trong các lớp học cụ thể. Các điều kiện ánh sáng và môi trường lớp học, vốn có thể thay đổi, sẽ được xem xét trong quá trình phát triển để đảm bảo hiệu suất tối ưu của hệ thống.

Nghiên cứu dự kiến kéo dài khoảng 3-4 tháng, bao gồm các giai đoạn quan trọng như thu thập dữ liệu, phát triển mô hình, và thử nghiệm hệ thống. Việc kiểm tra sẽ được thực hiện trong một kỳ học hoặc học kỳ cụ thể để đánh giá tính khả thi và hiệu quả trong điều kiện thực tế của lớp học.

Hệ thống sẽ sử dụng các công nghệ hiện đại như nhận diện khuôn mặt, học sâu (deep learning), và xử lý ảnh thông qua Python, với sự hỗ trợ của các thư viện như OpenCV, dlib, và TensorFlow/Keras. Mục tiêu là cải thiện quy trình điểm danh và quản lý danh sách sinh viên, góp phần nâng cao hiệu quả công tác điểm danh trong môi trường học tập.Để thực hiện đề tài, cần đảm bảo có các thiết bị cần thiết, bao gồm camera chất lượng tốt và máy tính đủ mạnh để xử lý dữ liệu. Dữ liệu khuôn mặt của sinh viên sẽ được thu thập trong môi trường học tập, và việc này sẽ được thực hiện theo đúng các quy định về quyền riêng tư và bảo mật thông tin.

Hệ thống có thể gặp một số thách thức như hiệu suất giảm trong điều kiện ánh sáng kém hoặc khi sinh viên đeo khẩu trang. Vì vậy, phạm vi nghiên cứu sẽ tập trung vào một nhóm sinh viên cụ thể trong một lớp học để kiểm tra tính khả thi và hiệu suất của hệ thống trong môi trường thực tế.

## **1.4 Đối tượng nghiên cứu**

Đề tài "Xây dựng hệ thống điểm danh sinh viên bằng khuôn mặt" tập trung nghiên cứu các đối tượng chính như sinh viên, giáo viên, hệ thống nhận diện khuôn mặt, thiết bị camera, cơ sở dữ liệu, phần mềm hệ thống, môi trường triển khai và bảo mật dữ liệu. Sinh viên là đối tượng chính được điểm danh, với các thông tin cần nghiên cứu bao gồm ảnh khuôn mặt từ nhiều góc độ, ánh sáng khác nhau, và thông tin cá nhân như mã sinh viên, họ tên, lớp học. Giáo viên đóng vai trò quản lý và theo dõi kết quả điểm danh, cần có giao diện trực quan để dễ dàng kiểm tra danh sách sinh viên có mặt. Hệ thống nhận diện khuôn mặt sử dụng các thuật toán như Haar Cascade, HOG hoặc các mô hình học sâu (CNN, YOLO), tích hợp thư viện hỗ trợ như OpenCV, Dlib hay FaceNet để đảm bảo khả năng nhận diện chính xác. Camera và thiết bị chụp ảnh cũng là yếu tố quan trọng, với yêu cầu nghiên cứu về độ phân giải, vị trí lắp đặt, và ảnh hưởng của môi trường như ánh sáng, góc quay.

Cơ sở dữ liệu giữ vai trò lưu trữ thông tin sinh viên, ảnh khuôn mặt và kết quả điểm danh, với yêu cầu thiết kế tối ưu, hỗ trợ truy xuất thời gian thực và tích hợp công nghệ lưu trữ SQL hoặc NoSQL. Phần mềm hệ thống cần có giao diện người dùng dễ sử dụng, phát triển bằng các công cụ như PyQt, Tkinter hoặc Flutter, để đảm bảo tính thân thiện và hiệu quả. Về môi trường triển khai, hệ thống cần được nghiên cứu xem nên hoạt động offline hay online, cũng như triển khai trên máy chủ cục bộ hay đám mây như AWS, Google Cloud hay XAMPP. Bên cạnh đó, bảo mật dữ liệu là yếu tố then chốt nhằm đảm bảo an toàn thông tin cá nhân và dữ liệu sinh viên, thông qua mã hóa và xây dựng quyền truy cập. Ngoài ra, để nâng cao hiệu quả, các thuật toán hỗ trợ như phát hiện khuôn mặt, giảm nhiễu, tối ưu hình ảnh, và đánh giá độ trễ cũng cần được nghiên cứu kỹ lưỡng. Việc tích hợp các yếu tố này sẽ đảm bảo hệ thống hoạt động hiệu quả, chính xác và đáp ứng tốt nhu cầu sử dụng thực tế.

## **1.5. Phương pháp nghiên cứu**

Trong đề tài "Xây dựng hệ thống điểm danh sinh viên bằng khuôn mặt", các phương pháp nghiên cứu được áp dụng nhằm đảm bảo tính khoa học và hiệu quả của hệ thống bao gồm:

### **1.5.1. Phương pháp thu thập thông tin**

Khảo sát: Tiến hành khảo sát nhu cầu sử dụng hệ thống từ phía giảng viên, sinh viên và các cán bộ quản lý để hiểu rõ yêu cầu thực tế về chức năng, độ chính xác, và tính tiện dụng.

Lập bảng hỏi: Thiết kế bảng hỏi để thu thập ý kiến về các yếu tố như giao diện hệ thống, mức độ tin cậy của nhận diện khuôn mặt, và các tình huống thực tế khi sử dụng.

Đọc tài liệu: Nghiên cứu các tài liệu học thuật, báo cáo kỹ thuật và các nghiên cứu trước đây liên quan đến nhận diện khuôn mặt, quản lý điểm danh và phát triển hệ thống thông tin. Các tài liệu này cung cấp cơ sở lý thuyết và các phương pháp triển khai phù hợp.

### **1.5.2. Phương pháp xử lý thông tin**

Phân tích định tính: Tổng hợp và phân tích thông tin thu thập được từ khảo sát, bảng hỏi và tài liệu để xác định các yêu cầu chức năng và phi chức năng của hệ thống. Phân tích định tính cũng được áp dụng để đánh giá mức độ hài lòng của người dùng sau khi thử nghiệm hệ thống.

Phân tích định lượng: Sử dụng dữ liệu từ khảo sát và thử nghiệm để tính toán các chỉ số như độ chính xác của nhận diện khuôn mặt, thời gian xử lý, và tỷ lệ lỗi. Kết quả định lượng giúp đánh giá hiệu năng và tối ưu hóa hệ thống.

### **1.5.3. Phương pháp thực nghiệm**

Thử nghiệm mô hình nhận diện khuôn mặt: Áp dụng các thuật toán nhận diện khuôn mặt khác nhau (Haar Cascade, HOG, CNN, YOLO) trên tập dữ liệu thu thập được để đánh giá và chọn ra phương pháp phù hợp nhất.

Kiểm tra hệ thống: Triển khai hệ thống thử nghiệm tại lớp học hoặc phòng thí nghiệm để đánh giá tính khả thi và hiệu quả. Các kết quả thử nghiệm thực tế sẽ được ghi nhận để điều chỉnh và cải tiến hệ thống.

### **1.5.4. Phương pháp thiết kế và mô phỏng**

Mô phỏng hệ thống: Thiết kế các mô hình mô phỏng để kiểm tra chức năng của hệ thống, từ việc phát hiện khuôn mặt đến ghi nhận thông tin điểm danh vào cơ sở dữ liệu.

Thiết kế giao diện: Sử dụng các công cụ như PyQt hoặc Tkinter để tạo ra giao diện trực quan, đáp ứng tốt nhu cầu của người dùng cuối.

### **1.5.5. Phương pháp thống kê**

Sử dụng các công cụ thống kê để phân tích dữ liệu thử nghiệm, đánh giá hiệu năng của hệ thống và đưa ra các biểu đồ minh họa như độ chính xác, tốc độ nhận diện, và tỷ lệ lỗi trong các trường hợp khác nhau.

### **1.5.6. Phương pháp so sánh và đánh giá**

So sánh hệ thống được xây dựng với các giải pháp nhận diện khuôn mặt hoặc điểm danh khác trên thị trường về các tiêu chí như độ chính xác, tốc độ, và chi phí triển khai.

### **1.5.7. Phương pháp chuyên gia**

Tìm ý kiến từ các chuyên gia trong lĩnh vực nhận diện khuôn mặt, xử lý ảnh và phát triển hệ thống thông tin để xác định các hướng tiếp cận phù hợp và khắc phục những khó khăn kỹ thuật.

## **1.6. Bố cục đề tài**

Chương 2 trình bày cơ sở lý thuyết và các nghiên cứu liên quan đến hệ thống điểm danh sinh viên bằng khuôn mặt. Nội dung chương này bao gồm tổng quan về các thuật toán nhận diện khuôn mặt phổ biến như Haar Cascade, HOG, và LBPH Recognizer, cùng với các đặc điểm nổi bật và ưu, nhược điểm của từng phương pháp. Ngoài ra, chương cũng đề cập đến các nghiên cứu trước đây về ứng dụng nhận diện khuôn mặt trong quản lý và điểm danh, qua đó phân tích những hạn chế và bài học kinh nghiệm từ các hệ thống hiện có. Những thông tin này đóng vai trò quan trọng trong việc định hướng phát triển hệ thống.

Trong Chương 3, báo cáo tập trung vào việc trình bày phương pháp luận và thiết kế hệ thống. Chương này giới thiệu quy trình xây dựng hệ thống, từ việc thu thập và tiền xử lý dữ liệu khuôn mặt, lựa chọn thuật toán nhận diện, đến việc thiết kế giao diện người dùng và cơ sở dữ liệu. Cấu trúc hệ thống được mô tả chi tiết, bao gồm các thành phần chính như module phát hiện khuôn mặt, module nhận diện, và module quản lý kết quả điểm danh. Ngoài ra, chương còn trình bày quy trình tích hợp các thành phần trên trong môi trường thử nghiệm, đảm bảo sự phối hợp nhịp nhàng giữa giao diện, cơ sở dữ liệu và camera.

# **CHƯƠNG 2. CƠ SỞ LÝ THUYẾT**

Trong đề tài này, hệ thống điểm danh sinh viên bằng khuôn mặt được xây dựng dựa trên sự kết hợp giữa thuật toán Haar Cascade và mô hình nhận diện LBPH (Local Binary Patterns Histograms). Thuật toán Haar Cascade được sử dụng để phát hiện nhanh vị trí khuôn mặt trong khung hình nhờ khả năng phân loại dựa trên các đặc điểm cường độ sáng tương phản. Sau khi phát hiện khuôn mặt, mô hình LBPH Recognizer được áp dụng để nhận diện khuôn mặt thông qua việc phân tích các mẫu đặc trưng cục bộ. Bên cạnh đó, PyQt6 được sử dụng làm bộ công cụ xây dựng giao diện đồ họa (GUI), cung cấp khả năng tạo các cửa sổ, nút bấm, hộp thông báo và nhiều thành phần giao diện khác để người dùng tương tác trực quan với hệ thống. Qt Designer, công cụ hỗ trợ thiết kế giao diện trực quan thông qua thao tác kéo thả, cho phép lưu giao diện dưới dạng file .ui, sau đó được nạp vào Python thông qua PyQt6 để hiển thị giao diện tương tác. Để quản lý cơ sở dữ liệu lưu trữ thông tin sinh viên và kết quả điểm danh, hệ thống sử dụng XAMPP với cơ sở dữ liệu MySQL. Python và các công nghệ nhận diện hiện đại khác cũng được tích hợp, đảm bảo tính linh hoạt và hiệu quả trong việc triển khai hệ thống điểm danh tự động.

## **2.1. PYTHON**

Python là một ngôn ngữ lập trình bậc cao, đa năng và rất phổ biến hiện nay. Nó được thiết kế với triết lý nhấn mạnh khả năng đọc mã, giúp các lập trình viên dễ dàng hiểu và viết code. Cú pháp của Python cũng rất rõ ràng và gần gũi với ngôn ngữ tự nhiên, làm giảm đáng kể độ phức tạp trong việc học và sử dụng.

### **2.1.1. Đơn giản, dễ học và dễ đọc**

Tính tối giản (Minimalistic): Python loại bỏ các yếu tố không cần thiết, chẳng hạn như dấu ngoặc nhọn {} hoặc dấu chấm phẩy ;, giúp mã nguồn trông gọn gàng và dễ duy trì.

Hỗ trợ mạnh cho người mới học: Ngoài cú pháp dễ hiểu, Python còn cung cấp nhiều công cụ như Jupyter Notebook, giúp học và thử nghiệm mã dễ dàng hơn.

### **2.1.2. Đa năng**

Khoa học dữ liệu và AI:Python là một trong những ngôn ngữ phổ biến nhất trong nghiên cứu AI nhờ sự tích hợp với các thư viện như TensorFlow, Keras và PyTorch.Thống kê và trực quan hóa cũng dễ dàng hơn với Matplotlib, Seaborn và Plotly.

Phát triển ứng dụng di động và GUI:Các công cụ như Kivy hoặc Tkinter cho phép bạn tạo ra ứng dụng di động hoặc giao diện người dùng đơn giản một cách nhanh chóng.

Xử lý hệ thống: Với os và subprocess, Python dễ dàng tương tác với hệ điều hành và thực hiện các tác vụ cấp hệ thống.

### **2.1.3. Mã nguồn mở**

Cộng đồng quốc tế: Python là một trong những ngôn ngữ có số lượng hội thảo, diễn đàn và tài nguyên trực tuyến phong phú nhất.

Được các tổ chức lớn ủng hộ: Google, NASA, Netflix, và nhiều công ty hàng đầu đều sử dụng Python trong các dự án quan trọng.

### **2.1.4. Thư viện phong phú**

Tài chính: Python cung cấp các thư viện như QuantLib, Zipline, và PyAlgoTrade để hỗ trợ phân tích tài chính và giao dịch tự động.

Sinh học và y học: Biopython và PySCeS hỗ trợ xử lý dữ liệu sinh học.

Xử lý âm thanh và video: MoviePy, OpenCV, Pydub giúp làm việc với âm thanh và video.

Hỗ trợ Machine Learning & Deep Learning: Bên cạnh các thư viện nổi tiếng, Python còn tích hợp với Google Colab, một môi trường chạy mã trực tuyến miễn phí với GPU.

### **2.1.5. Di động**

Python trên nền tảng di động: Framework như BeeWare hoặc Chaquopy cho phép phát triển ứng dụng trên Android và iOS bằng Python.

Khả năng tích hợp mạnh mẽ: Python có thể được sử dụng để viết script hoặc plugin trong các ứng dụng như Blender, GIMP, và AutoCAD.

### **2.1.6. Hướng đối tượng**

Hỗ trợ nhiều phong cách lập trình: Python không chỉ hỗ trợ lập trình hướng đối tượng (OOP) mà còn lập trình hàm (functional) và thủ tục (procedural), giúp linh hoạt trong việc xây dựng ứng dụng.

### **2.1.7. Tính tương tác**

Notebook-based Development: Các công cụ như Jupyter Notebook cho phép viết mã, ghi chú, và trực quan hóa dữ liệu trong cùng một giao diện, rất phù hợp với giáo dục và nghiên cứu.

### **2.1.8. Tính bảo mật và khả năng mở rộng**

Bảo mật:Python cung cấp các thư viện như Cryptography, PyJWT, và hashlib để mã hóa dữ liệu và xác thực người dùng.

Mở rộng:Python có thể tích hợp với các ngôn ngữ khác như C, C++, hoặc Java thông qua các công cụ như SWIG, ctypes, hoặc Jython.

### **2.1.8. Ưu điểm của Python**

Dễ học và dễ sử dụng:Cú pháp đơn giản, giống ngôn ngữ tự nhiên, giúp người mới dễ dàng tiếp cận.Không yêu cầu khai báo kiểu dữ liệu, giảm thời gian viết mã.

Đa năng và linh hoạt:Có thể sử dụng cho nhiều lĩnh vực như phát triển web, khoa học dữ liệu, trí tuệ nhân tạo, tự động hóa, và nhiều hơn nữa.Chạy được trên nhiều nền tảng mà không cần sửa đổi mã nguồn (Windows, macOS, Linux, v.v.).

Thư viện phong phú:Python cung cấp hàng ngàn thư viện và framework, từ thư viện tiêu chuẩn đến các thư viện của bên thứ ba, giúp lập trình viên không cần xây dựng mọi thứ từ đầu.

Cộng đồng lớn và hỗ trợ mạnh mẽ:Cộng đồng người dùng Python đông đảo, dễ dàng tìm kiếm tài liệu, khóa học và nhận hỗ trợ từ diễn đàn trực tuyến.

Tích hợp và mở rộng dễ dàng:Có thể tích hợp với các ngôn ngữ khác như C, C++, Java hoặc sử dụng Python để viết plugin hoặc script trong các phần mềm khác.

Tính di động:Mã Python có thể chạy trên bất kỳ hệ điều hành nào miễn là đã cài Python Interpreter, không cần biên dịch lại mã nguồn.

Hỗ trợ lập trình hướng đối tượng:Python cho phép tổ chức mã nguồn khoa học và có cấu trúc nhờ vào việc hỗ trợ lập trình hướng đối tượng (Object-Oriented Programming).

### **2.1.9. Nhược điểm của Python**

Hiệu năng không cao:Python là ngôn ngữ thông dịch (interpreted language), dẫn đến tốc độ chạy chương trình thường chậm hơn so với các ngôn ngữ biên dịch như C++ hoặc Java.

Không phù hợp cho ứng dụng yêu cầu thời gian thực:Với những ứng dụng cần hiệu suất cao hoặc yêu cầu thời gian thực như game 3D phức tạp hoặc các hệ thống nhúng, Python có thể không phải là lựa chọn tối ưu.

Tiêu thụ tài nguyên lớn:Vì Python không tối ưu hóa hiệu suất, nó thường tiêu tốn nhiều RAM hơn các ngôn ngữ khác khi xử lý khối lượng công việc lớn.

Giới hạn trong di động và phát triển ứng dụng di động:Python không phải là lựa chọn hàng đầu để phát triển ứng dụng di động trên iOS hoặc Android, dù có các framework như Kivy hay BeeWare nhưng chúng chưa phổ biến và không mạnh mẽ như các công cụ chuyên dụng khác.

Khó bảo trì mã nguồn trong các dự án lớn:Do không cần khai báo kiểu dữ liệu, Python có thể dẫn đến lỗi tiềm ẩn trong các dự án phức tạp khi phải làm việc với nhiều lập trình viên.

Hạn chế trong lập trình đa luồng:Global Interpreter Lock (GIL) của Python ngăn cản thực thi đa luồng (multi-threading) hiệu quả, làm giảm hiệu suất trong các tác vụ tính toán nặng.

Phụ thuộc vào Interpreter:Python cần có môi trường Python Interpreter để chạy, điều này có thể gây khó khăn khi triển khai ứng dụng trong một số môi trường cụ thể.

## **2.2. XAMP**

### **2.2.1. Xampp là gì và công dụng**

XAMPP là một phần mềm mã nguồn mở được thiết kế để tạo môi trường máy chủ cục bộ (localhost) trên máy tính cá nhân. Tên gọi XAMPP là viết tắt của các thành phần chính: Cross-platform (X), Apache (A), MariaDB/MySQL (M), PHP (P), và Perl (P). Đây là một công cụ hữu ích, giúp lập trình viên phát triển và thử nghiệm các ứng dụng web trong môi trường giả lập trước khi triển khai lên máy chủ thực tế. Với XAMPP, người dùng có thể dễ dàng thiết lập máy chủ ngay trên máy tính mà không cần kết nối internet hay sử dụng máy chủ vật lý. Phần mềm này hỗ trợ chạy các ứng dụng web, quản lý cơ sở dữ liệu, thử nghiệm các tính năng mã nguồn, tích hợp API, và thậm chí hỗ trợ cài đặt các hệ quản trị nội dung phổ biến như WordPress hay Joomla.

### **2.2.2. Chức năng của xampp**

XAMPP cung cấp một môi trường lý tưởng cho lập trình viên bằng cách tích hợp các dịch vụ quan trọng như Apache, MariaDB, PHP, và phpMyAdmin. Apache hoạt động như máy chủ web chính, chịu trách nhiệm xử lý và cung cấp nội dung từ ứng dụng web đến trình duyệt. MariaDB, trước đây là MySQL, đóng vai trò là hệ quản trị cơ sở dữ liệu, giúp lưu trữ và quản lý thông tin. Thông qua phpMyAdmin, người dùng có thể tương tác với cơ sở dữ liệu thông qua giao diện trực quan. Ngoài ra, XAMPP còn hỗ trợ các ngôn ngữ lập trình như PHP và Perl, đồng thời cung cấp các dịch vụ bổ sung như FileZilla FTP Server và Mercury Mail Server để quản lý tệp và email trong môi trường thử nghiệm. Nhờ đó, người dùng có thể thực hiện từ việc phát triển ứng dụng đến kiểm thử các tính năng phức tạp một cách dễ dàng.

### **2.2.3. Ưu điểm và nhược điểm của xampp**

XAMPP nổi bật với tính dễ cài đặt và sử dụng, phù hợp cho cả người mới bắt đầu lẫn lập trình viên có kinh nghiệm. Phần mềm này hoàn toàn miễn phí, mã nguồn mở, và hoạt động trên nhiều hệ điều hành như Windows, macOS, và Linux, giúp tăng tính linh hoạt trong phát triển. XAMPP tích hợp sẵn các công cụ cần thiết, tiết kiệm thời gian và công sức cho người dùng. Đồng thời, cộng đồng lớn mạnh của XAMPP cũng mang lại nhiều tài liệu và hỗ trợ hữu ích.

Tuy nhiên, XAMPP cũng có một số hạn chế. Đây là công cụ được thiết kế cho môi trường phát triển và thử nghiệm, nên không được tối ưu cho máy chủ sản xuất. Bên cạnh đó, XAMPP có mức độ bảo mật thấp khi cấu hình mặc định, điều này có thể gây rủi ro nếu sử dụng trong môi trường công khai. Ngoài ra, phần mềm này tiêu tốn khá nhiều tài nguyên khi bật nhiều dịch vụ đồng thời, đặc biệt trên các máy tính có cấu hình thấp.

## **2.3. QT DESIGNER**

### **2.3.1. Qt Designer là gì**

Qt Designer là một công cụ mạnh mẽ được phát triển bởi Qt Company, được sử dụng để thiết kế giao diện đồ họa người dùng (GUI - Graphical User Interface) cho các ứng dụng. Công cụ này là một phần của Qt Framework, cung cấp khả năng kéo thả (drag-and-drop) trực quan, giúp người dùng dễ dàng tạo ra các giao diện phức tạp mà không cần viết mã nguồn từ đầu. Qt Designer hỗ trợ thiết kế giao diện cho nhiều nền tảng (cross-platform) như Windows, macOS, Linux, và có thể được sử dụng với nhiều ngôn ngữ lập trình, phổ biến nhất là Python (thông qua PyQt hoặc PySide)

### **2.3.2. Chức năng của Qt Designer**

Qt Designer cho phép người dùng tạo các thành phần giao diện như cửa sổ chính, hộp thoại, thanh công cụ, menu, và nhiều loại widget khác một cách trực quan. Sau khi thiết kế, giao diện sẽ được lưu dưới dạng file XML với định dạng .ui (User Interface). File này có thể được chuyển đổi thành mã nguồn (Python hoặc C++) bằng các công cụ hỗ trợ như pyuic (cho Python) hoặc sử dụng trực tiếp trong ứng dụng.

Một số chức năng chính của Qt Designer bao gồm:

Tạo giao diện bằng phương pháp kéo thả:Người dùng có thể chọn và kéo các widget (nút bấm, hộp văn bản, thanh trượt, v.v.) từ thư viện widget của Qt Designer vào giao diện thiết kế.

Tùy chỉnh thuộc tính:Cung cấp bảng thuộc tính để thay đổi các cài đặt như màu sắc, kích thước, font chữ, và hành vi của các widget.

Hỗ trợ bố cục giao diện (Layout):Qt Designer giúp sắp xếp widget theo các bố cục như lưới (grid), hàng ngang (horizontal), hoặc hàng dọc (vertical) để đảm bảo giao diện linh hoạt và thân thiện với người dùng.

Hỗ trợ thiết kế đa nền tảng:Giao diện được tạo bởi Qt Designer có thể chạy trên nhiều hệ điều hành mà không cần thay đổi mã nguồn.

Tích hợp với mã nguồn:Các file .ui có thể được chuyển đổi thành mã nguồn hoặc sử dụng trực tiếp để liên kết với logic xử lý của ứng dụng.

### **2.3.3. Công dụng của Qt Designer**

Thiết kế giao diện đồ họa:Qt Designer cho phép lập trình viên nhanh chóng tạo giao diện trực quan mà không cần tốn quá nhiều thời gian viết mã.

Tăng hiệu suất phát triển:Công cụ này giảm bớt khối lượng công việc liên quan đến việc xây dựng giao diện bằng mã nguồn, giúp lập trình viên tập trung hơn vào logic xử lý của ứng dụng.

Tích hợp dễ dàng:Giao diện được thiết kế trong Qt Designer có thể dễ dàng tích hợp với các ngôn ngữ lập trình như Python (PyQt/PySide) hoặc C++.

Thân thiện với người mới bắt đầu:Với giao diện kéo thả và bảng thuộc tính trực quan, người dùng mới làm quen với lập trình GUI có thể dễ dàng tiếp cận.

### **2.3.4. Ưu điểm và nhược điểm của Qt Designer**

Ưu điểm

Trực quan và dễ sử dụng:Qt Designer giúp người dùng nhanh chóng tạo giao diện mà không cần viết mã.

Hỗ trợ đa nền tảng:Giao diện thiết kế có thể hoạt động trên nhiều hệ điều hành.

Tích hợp mạnh mẽ:Dễ dàng kết nối giao diện với logic xử lý trong Python hoặc C++.

Tài liệu phong phú và cộng đồng lớn:Qt Designer có nhiều tài liệu hỗ trợ và cộng đồng đông đảo sẵn sàng giúp đỡ.

Nhược điểm

Phụ thuộc vào Qt Framework:Người dùng cần học cách sử dụng Qt Framework để tận dụng hết các tính năng của Qt Designer.

Thiếu tính tùy chỉnh chi tiết trong một số trường hợp:Mặc dù giao diện kéo thả rất tiện lợi, nhưng đôi khi cần viết mã thủ công để đạt được các tính năng phức tạp hơn.

## **2.4. Thuật toán Haar-cascade**

### **2.4.1. Thuật toán Haar Cascade là gì?**

Thuật toán Haar Cascade là một phương pháp nhận diện đối tượng nổi bật, đặc biệt được sử dụng trong nhận diện khuôn mặt, được đề xuất lần đầu bởi Paul Viola và Michael Jones vào năm 2001. Thuật toán này là một phần quan trọng trong nhận diện hình ảnh và được áp dụng rộng rãi trong các ứng dụng nhận diện khuôn mặt, xe cộ, và nhiều đối tượng khác trong các hệ thống xử lý ảnh.

Haar Cascade dựa trên việc sử dụng các đặc trưng Haar-like (tương tự như bộ lọc, được sử dụng để nhận diện các đặc điểm trong hình ảnh) để phát hiện các đặc điểm quan trọng của một đối tượng trong ảnh, sau đó sử dụng một bộ phân loại Cascade để phân loại các vùng trong ảnh là đối tượng hay không phải là đối tượng.

### **2.4.2. Cấu trúc và nguyên lý hoạt động của thuật toán Haar Cascade**

Thuật toán Haar Cascade hoạt động qua các bước chính như sau:

#### **2.4.2.1. Haar Features (đặc trưng Haar)**

Đặc trưng Haar là các hình chữ nhật màu đen và trắng được đặt trong ảnh để làm nổi bật các đặc điểm trong ảnh. Những đặc trưng này giúp xác định sự khác biệt giữa các vùng trong ảnh (ví dụ, sự khác biệt giữa nền và đối tượng).

Các đặc trưng Haar giống như bộ lọc có thể phát hiện các tính năng như cạnh, đường viền, và các đối tượng hình chữ nhật đơn giản. Chúng có thể được tính toán rất nhanh chóng và hiệu quả.

#### **2.4.2.2. Integral Image (hình ảnh tổng hợp)**

Để tính toán các đặc trưng Haar một cách nhanh chóng, thuật toán sử dụng khái niệm "hình ảnh tổng hợp". Một hình ảnh tổng hợp là một phiên bản của ảnh gốc mà trong đó mỗi điểm ảnh chứa tổng của tất cả các pixel phía trên và bên trái của nó.

Điều này giúp giảm thiểu việc tính toán lại tổng số pixel mỗi khi di chuyển một cửa sổ quét qua ảnh.

#### **2.4.2.3. Cascade Classifier (bộ phân loại Cascade)**

Sau khi tính toán các đặc trưng Haar, thuật toán sử dụng một bộ phân loại Cascade để phân loại các vùng ảnh thành "đối tượng" (ví dụ như khuôn mặt) hoặc "không phải đối tượng".

Cascade Classifier là một loạt các bộ phân loại yếu, mỗi bộ phân loại được huấn luyện để nhận diện các đặc trưng cụ thể của đối tượng. Các bộ phân loại này được sắp xếp theo một cấu trúc dạng Cascade, với các bộ phân loại đơn giản ở đầu (dễ tính toán) và các bộ phân loại phức tạp hơn ở cuối.

Việc sắp xếp này giúp giảm thiểu chi phí tính toán, vì nếu một bộ phân loại đầu tiên xác định rõ ràng không phải là đối tượng, nó sẽ bỏ qua các bộ phân loại tiếp theo.

#### **2.4.2.4. Huấn luyện bộ phân loại**

Quá trình huấn luyện bộ phân loại Haar Cascade sử dụng một phương pháp gọi là AdaBoost (Adaptive Boosting). AdaBoost kết hợp các bộ phân loại yếu thành một bộ phân loại mạnh, giúp tăng độ chính xác của nhận diện.

#### **2.4.2.5. Quá trình quét (Sliding Window)**

Thuật toán sử dụng phương pháp quét cửa sổ (sliding window) để kiểm tra từng phần của hình ảnh, từ đó phát hiện các đối tượng. Cửa sổ sẽ di chuyển qua ảnh và kiểm tra các đặc trưng Haar trong mỗi phần của ảnh.

### **2.4.3. Ưu điểm và nhược điểm của Haar Cascade**

Ưu điểm

Nhanh và hiệu quả:Haar Cascade có thể chạy nhanh nhờ vào việc sử dụng đặc trưng Haar và hình ảnh tổng hợp, giúp quá trình tính toán được tối ưu hóa.

Dễ triển khai:Thuật toán dễ dàng triển khai với các thư viện như OpenCV, điều này giúp lập trình viên nhanh chóng áp dụng vào các dự án nhận diện đối tượng.

Được huấn luyện sẵn:OpenCV cung cấp các mô hình Haar Cascade đã được huấn luyện sẵn cho các đối tượng như khuôn mặt, mắt, và xe hơi, giúp tiết kiệm thời gian huấn luyện.

Nhược điểm

Chỉ phù hợp với các đối tượng có đặc điểm rõ ràng:Haar Cascade hoạt động tốt nhất khi đối tượng có đặc điểm dễ phân biệt với nền, nhưng đối với các đối tượng phức tạp hoặc có sự thay đổi lớn về hình dáng và kích thước, thuật toán có thể gặp khó khăn.

Không mạnh mẽ trong điều kiện ánh sáng và góc nhìn thay đổi:Thuật toán Haar Cascade có thể không hoạt động tốt trong các điều kiện ánh sáng kém hoặc khi đối tượng bị che khuất một phần, hoặc trong các góc nhìn không chuẩn.

Khả năng nhận diện giới hạn:Haar Cascade có thể gặp khó khăn khi phải xử lý các đối tượng có sự biến động lớn về màu sắc, hình dạng hay kích thước trong các bối cảnh phức tạp.

## **2.5. Mô hình Nhận diện LBPH Recognizer**

LBPH Recognizer (Local Binary Patterns Histogram Recognizer) là một mô hình được sử dụng để nhận diện khuôn mặt trong các hệ thống nhận diện khuôn mặt. Thuật toán này dựa trên đặc trưng Local Binary Pattern (LBP), một phương pháp trích xuất đặc trưng hình ảnh mạnh mẽ và phổ biến, đặc biệt trong việc nhận diện khuôn mặt. Mô hình nhận diện LBPH có thể xác định và nhận diện các khuôn mặt từ các ảnh đã được huấn luyện trước.

### **2.5.1. Tổng quan về LBPH (Local Binary Patterns Histogram)**

Thuật toán LBPH hoạt động dựa trên một đặc trưng hình ảnh gọi là Local Binary Patterns (LBP). Đặc trưng này được tính toán bằng cách so sánh giá trị các pixel trong một cửa sổ nhỏ với giá trị pixel trung tâm, tạo ra một mã nhị phân cho mỗi cửa sổ.

Quá trình tính toán LBP:

Mỗi pixel trong một khu vực được so sánh với pixel trung tâm của khu vực đó. Nếu pixel trong vùng lân cận lớn hơn hoặc bằng pixel trung tâm, giá trị của pixel đó sẽ được gán là 1; nếu không, giá trị sẽ là 0.

Mỗi cửa sổ tạo ra một mã nhị phân, và mã nhị phân này được chuyển đổi thành giá trị thập phân.

Sau khi áp dụng LBP cho toàn bộ ảnh, một histogram (tần suất phân phối các mã LBP) sẽ được tạo ra, đại diện cho đặc trưng của khuôn mặt hoặc đối tượng trong ảnh.

### **2.5.2. Quá trình nhận diện với LBPH Recognizer**

Quá trình nhận diện của LBPH Recognizer có thể được chia thành hai giai đoạn chính: huấn luyện và nhận diện.

Giai đoạn huấn luyện:Thu thập dữ liệu ảnh khuôn mặt:Dữ liệu huấn luyện bao gồm ảnh của các khuôn mặt cùng với nhãn của từng khuôn mặt (ví dụ: tên hoặc ID của người trong ảnh).

Trích xuất đặc trưng LBP:Đối với mỗi ảnh khuôn mặt trong cơ sở dữ liệu, thuật toán LBP sẽ được áp dụng để trích xuất các đặc trưng. Mỗi khuôn mặt sẽ có một histogram LBP riêng biệt.

Lưu trữ các đặc trưng:Các histogram LBP sẽ được lưu trữ cùng với nhãn của khuôn mặt. Những đặc trưng này sẽ được sử dụng để huấn luyện mô hình nhận diện.

Sử dụng một bộ phân loại (Classifier):Thường thì LBPH Recognizer sẽ sử dụng K-nearest neighbors (k-NN) làm bộ phân loại để phân loại các histogram và xác định khuôn mặt tương ứng.

Giai đoạn nhận diện:

Nhận diện khuôn mặt mới:Khi nhận diện khuôn mặt trong ảnh mới, hệ thống đầu tiên phát hiện khuôn mặt trong ảnh (thường sử dụng Haar-Cascade hoặc các phương pháp phát hiện khuôn mặt khác).

Trích xuất đặc trưng LBP:Tương tự như trong quá trình huấn luyện, hệ thống sẽ áp dụng LBP để trích xuất các đặc trưng từ khuôn mặt mới.

So sánh với cơ sở dữ liệu:Histogram LBP của khuôn mặt mới sẽ được so sánh với các histogram trong cơ sở dữ liệu đã huấn luyện. Hệ thống sẽ tìm kiếm histogram gần nhất và trả về nhãn của khuôn mặt tương ứng.

Phân loại:Sau khi so sánh, LBPH Recognizer sẽ phân loại khuôn mặt trong ảnh và trả về kết quả nhận diện, ví dụ như tên của người có khuôn mặt tương ứng.

## **2.6. Thuật toán LBPH (Local Binary Patterns Histograms)**

LBPH (Local Binary Patterns Histograms) là một thuật toán mạnh mẽ và phổ biến trong lĩnh vực nhận diện khuôn mặt và xử lý hình ảnh. Thuật toán này sử dụng phương pháp trích xuất đặc trưng từ các bức ảnh bằng cách sử dụng các mẫu nhị phân (binary patterns) để mô tả đặc điểm hình ảnh một cách chính xác. Đây là một trong những phương pháp hiệu quả nhất trong việc nhận diện khuôn mặt trong các điều kiện ánh sáng và góc độ thay đổi.

Quy trình hoạt động của LBPH trong nhận diện khuôn mặt bao gồm các bước cơ bản:

Tiền xử lý ảnh: Ảnh được chuyển sang ảnh xám và chia thành các vùng nhỏ.

Tính toán LBP: LBP được tính cho mỗi vùng nhỏ và chuyển thành mã nhị phân.

Xây dựng histogram: Các mã LBP được tổng hợp thành histogram, đại diện cho đặc trưng của khuôn mặt.

Nhận diện khuôn mặt: So sánh histogram của khuôn mặt mới với các histogram đã lưu trữ từ các khuôn mặt đã được huấn luyện trước đó.

Thuật toán LBPH có ưu điểm là dễ cài đặt, yêu cầu ít tài nguyên tính toán và có khả năng nhận diện khuôn mặt trong các điều kiện ánh sáng thay đổi. Tuy nhiên, nó có thể gặp khó khăn khi nhận diện khuôn mặt trong các tình huống có sự thay đổi góc nhìn hoặc biểu cảm khuôn mặt.

# **CHƯƠNG 3. PHÂN TÍCH HỆ THỐNG VÀ XÂY DỰNG SẢN PHẨM**

Trong chương này, chúng tôi sẽ trình bày chi tiết quá trình xây dựng sản phẩm và mô hình hệ thống cho đề tài "Hệ thống điểm danh sinh viên bằng khuôn mặt". Mục tiêu của chương này là mô tả các bước thực hiện từ phân tích yêu cầu hệ thống đến việc thiết kế, phát triển và triển khai ứng dụng thực tế.

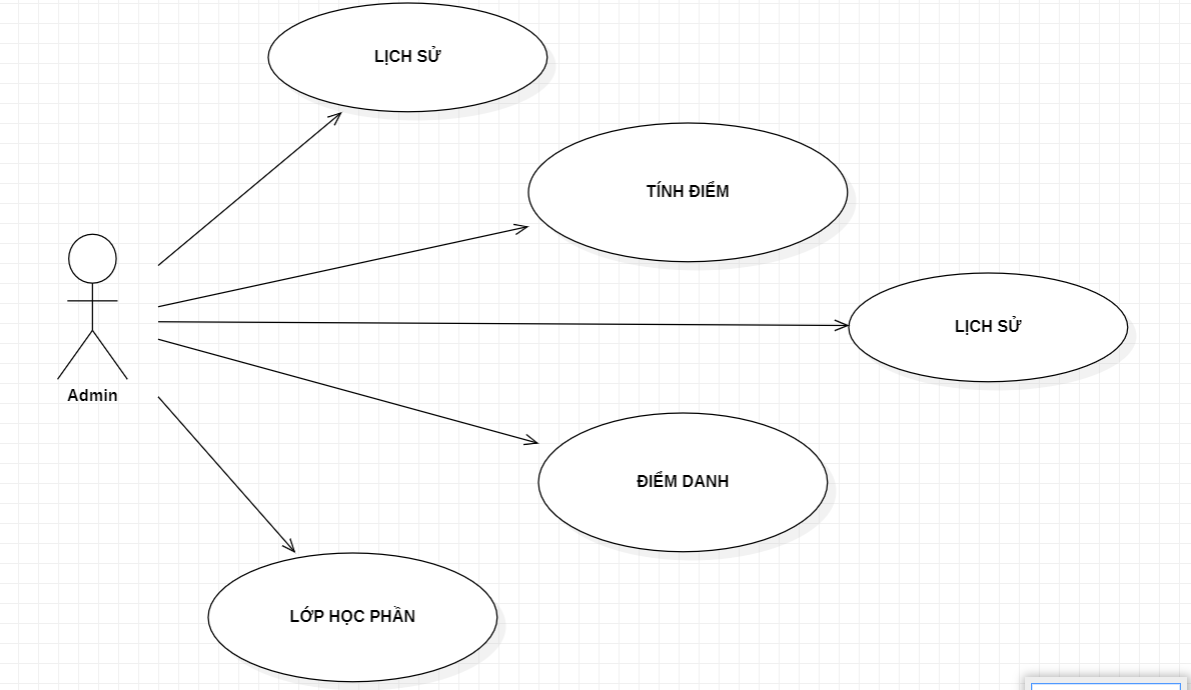
Sản phẩm của hệ thống nhằm tự động hóa quá trình điểm danh sinh viên trong các lớp học thông qua công nghệ nhận diện khuôn mặt. Hệ thống sẽ giúp giảng viên tiết kiệm thời gian trong việc kiểm tra điểm danh, giảm thiểu các lỗi do việc ghi chép thủ công, đồng thời tạo ra một giải pháp an toàn và tiện lợi hơn cho các cơ sở giáo dục. Hệ thống sẽ hoạt động bằng cách nhận diện khuôn mặt của sinh viên qua camera, và tự động ghi nhận thời gian điểm danh, đồng thời lưu trữ dữ liệu vào cơ sở dữ liệu cho việc thống kê và báo cáo sau này.

Mô hình hệ thống được thiết kế theo kiến trúc client-server, trong đó các thành phần chính bao gồm giao diện người dùng, mô-đun nhận diện khuôn mặt, cơ sở dữ liệu lưu trữ thông tin sinh viên và điểm danh, và mô-đun xử lý báo cáo. Giao diện người dùng được phát triển với PyQt6 để tạo ra môi trường dễ sử dụng và thân thiện cho giảng viên. Phần nhận diện khuôn mặt sử dụng các thư viện mạnh mẽ như OpenCV và Haar Cascade để phát hiện và nhận diện khuôn mặt từ ảnh chụp hoặc video trực tiếp. Thông tin về điểm danh sẽ được lưu trữ và quản lý qua MySQL, đảm bảo tính ổn định và bảo mật cho dữ liệu người dùng

## **3.1. Phân tích hệ thống**

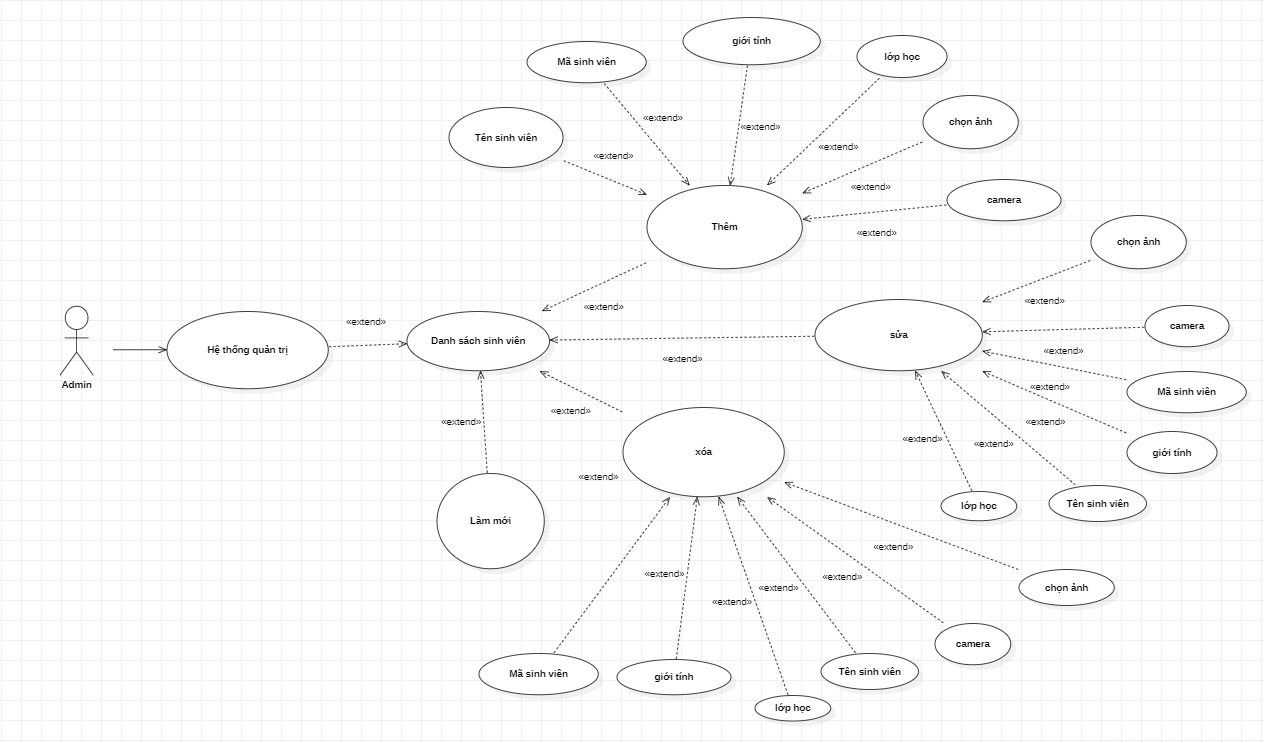
### **3.1.1 Sơ Đồ Usecase**

Sơ đồ use case tổng quát của hệ thống nhận diện sinh viên

****

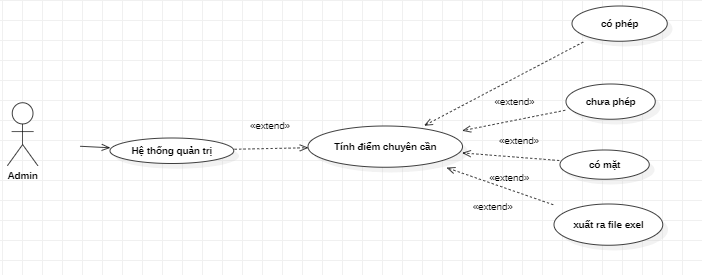
Hình 3.1. Usecase tổng quát

Sơ đồ use case ở chức năng sinh viên

****

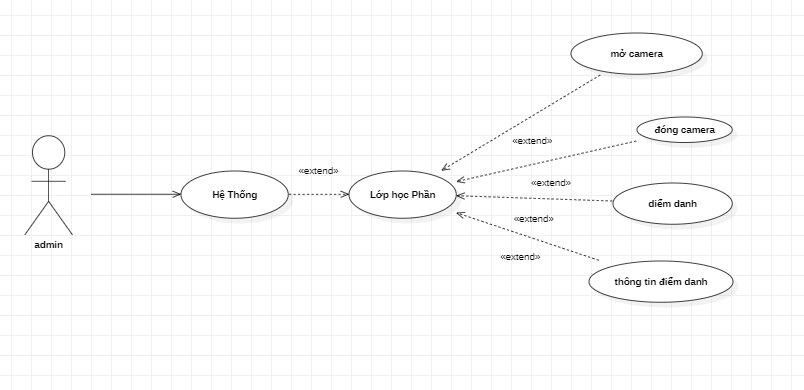
Hình 3.2. Usecase sinh viên

Sơ đồ use case ở chức năng tính điểm chuyên cần



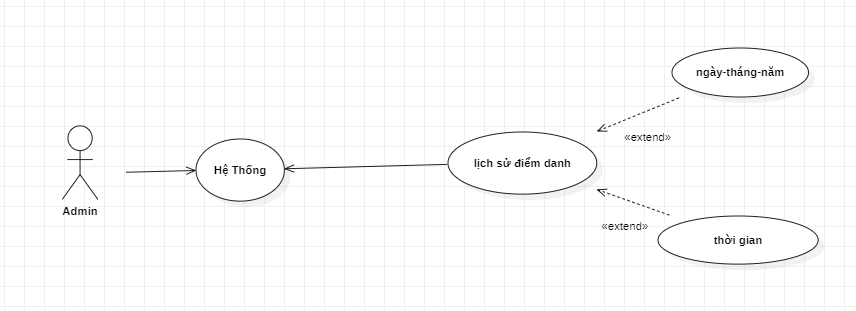
Hình 3.3. Usecase tính điểm chuyên cần

Sơ đồ use case ở chức năng lớp học phần

****

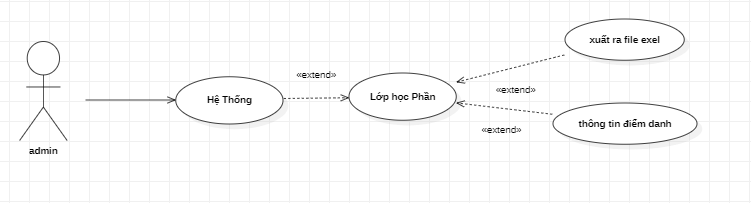
Hình 3.4. Usecase lớp học phần

Sơ đồ use case ở chức năng lịch sử điểm danh

****

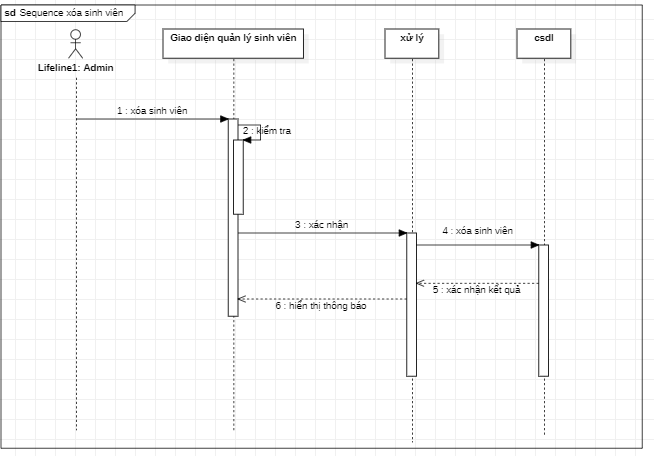
Hình 3.5. Usecase lịch sử điểm danh

Sơ đồ use case ở chức năng lớp học phần

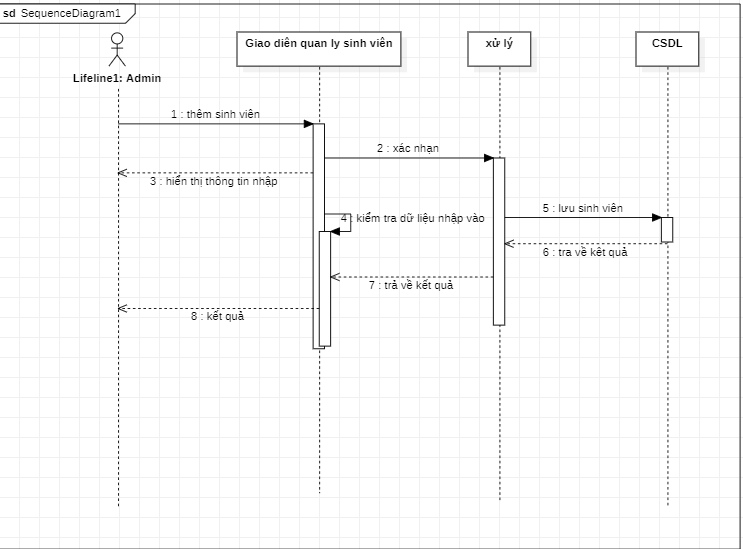
****

Hình 3.6. Usecase lớp học phần

### **3.1.2. Sơ Đồ Tuần Tự**

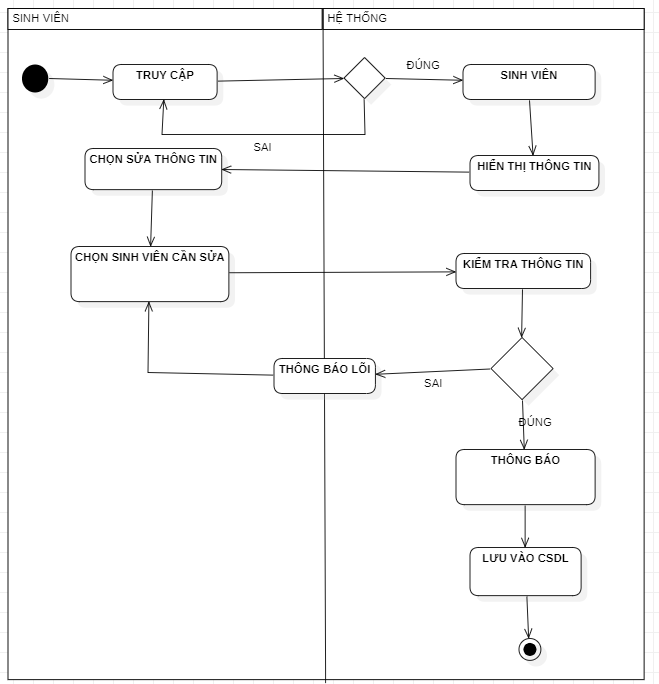
****

Hình 3.7. Sơ đồ tuần tự xóa sinh viên

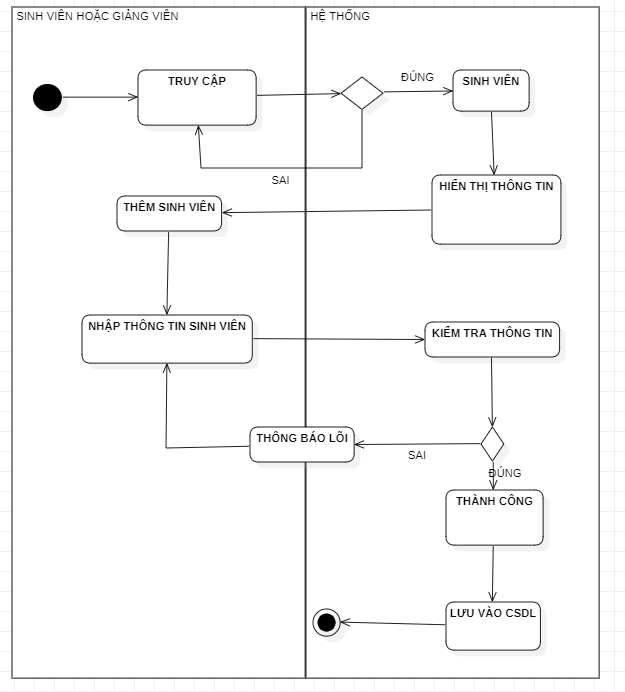
****

Hình 3.8. Sơ đồ tuần tự thêm sinh viên

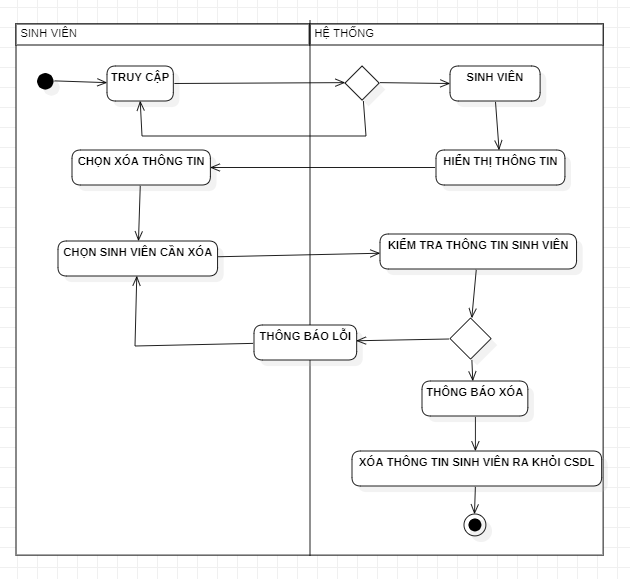
### **3.1.3. Sơ Đồ Hoạt Động**

****

Hình 3.9. Sơ đồ hoạt động của cập nhật sinh viên

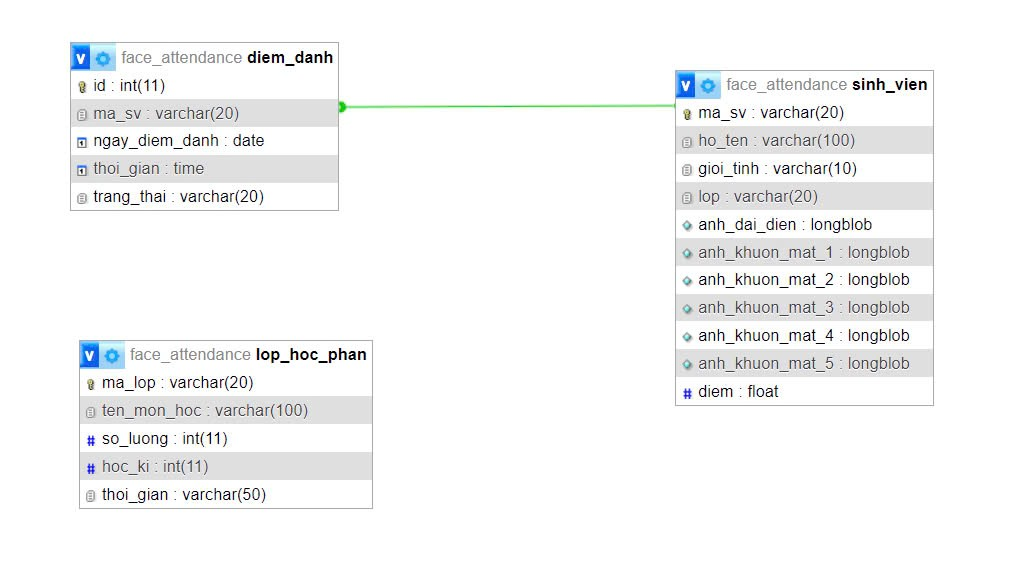
****

Hình 3.10. Sơ đồ hoạt động của thêm sinh viên

****

Hình 3.11. Sơ đồ hoạt động của xóa sinh viên

**3.1.4. Database**

****

Hình 3.12. Database

## **3.2. Xây dựng giao diện sản phẩm**

### **3.2.1 Chức Năng Sản Phẩm**

**3.2.1.1. Chức năng quản lý thông tin sinh viên**

Thêm sinh viên:Nhập thông tin cá nhân bao gồm: mã số sinh viên (MSSV), tên, lớp, giới tính và hình ảnh khuôn mặt.

Xóa sinh viên:Xóa hoàn toàn thông tin sinh viên không còn sử dụng trong hệ thống.

Sửa thông tin sinh viên:Cho phép cập nhật lại các thông tin như tên, lớp, hình ảnh.

Làm mới danh sách:Tự động hiển thị lại toàn bộ danh sách sinh viên hiện tại sau khi có thay đổi.

Nhập liệu:Tải thông tin sinh viên từ file Excel để thêm nhanh vào hệ thống.

**3.2.1.2. Chức năng điểm danh**

Nhận diện khuôn mặt:Sử dụng công nghệ nhận diện khuôn mặt để xác định danh tính sinh viên khi vào lớp.So khớp khuôn mặt với dữ liệu có sẵn trong cơ sở dữ liệu.

Điểm danh trực tiếp:Ghi nhận thời gian điểm danh chính xác, trạng thái điểm danh (có mặt).

Cập nhật thông tin điểm danh:Lưu trạng thái điểm danh (đã điểm danh, chưa điểm danh) và thời gian vào hệ thống theo từng buổi học.

**3.2.1.3. Chức năng quản lý lớp học phần**

Hiển thị danh sách lớp học phần:Danh sách mã lớp học phần và tên lớp được hiển thị đầy đủ trên giao diện quản trị.

Thêm lớp học phần:Thêm mới thông tin mã lớp học phần vào hệ thống.

Xuất danh sách sinh viên:Xuất file Excel chứa danh sách sinh viên trong từng lớp học phần, bao gồm thông tin điểm danh.

**3.2.1.4. Chức năng tính điểm**

Tính điểm quá trình:Tự động tính điểm dựa trên trạng thái điểm danh của sinh viên:

Có phép (vắng mặt có lý do hợp lệ)

Không phép (vắng mặt không có lý do)

Có mặt (tham gia đầy đủ buổi học)

Điểm quá trình được tính dựa trên tỷ lệ buổi có mặt và quy tắc đánh giá của giảng viên.

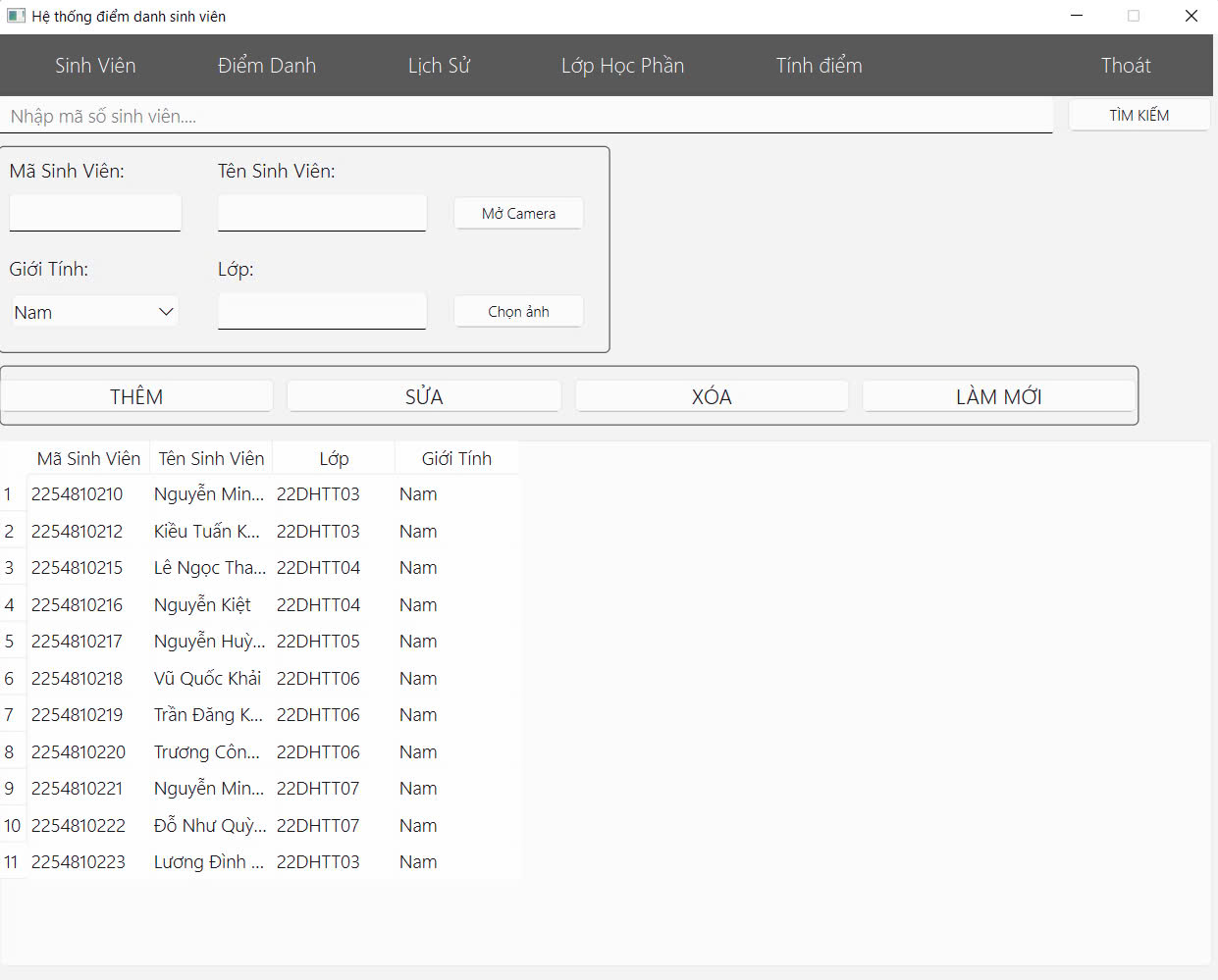
**3.2.1.5. Chức năng lịch sử điểm danh**

Xem lịch sử:Hiển thị danh sách chi tiết sinh viên đã điểm danh theo mã số sinh viên và tên sinh viên và ngày và thời gian điểm danh.

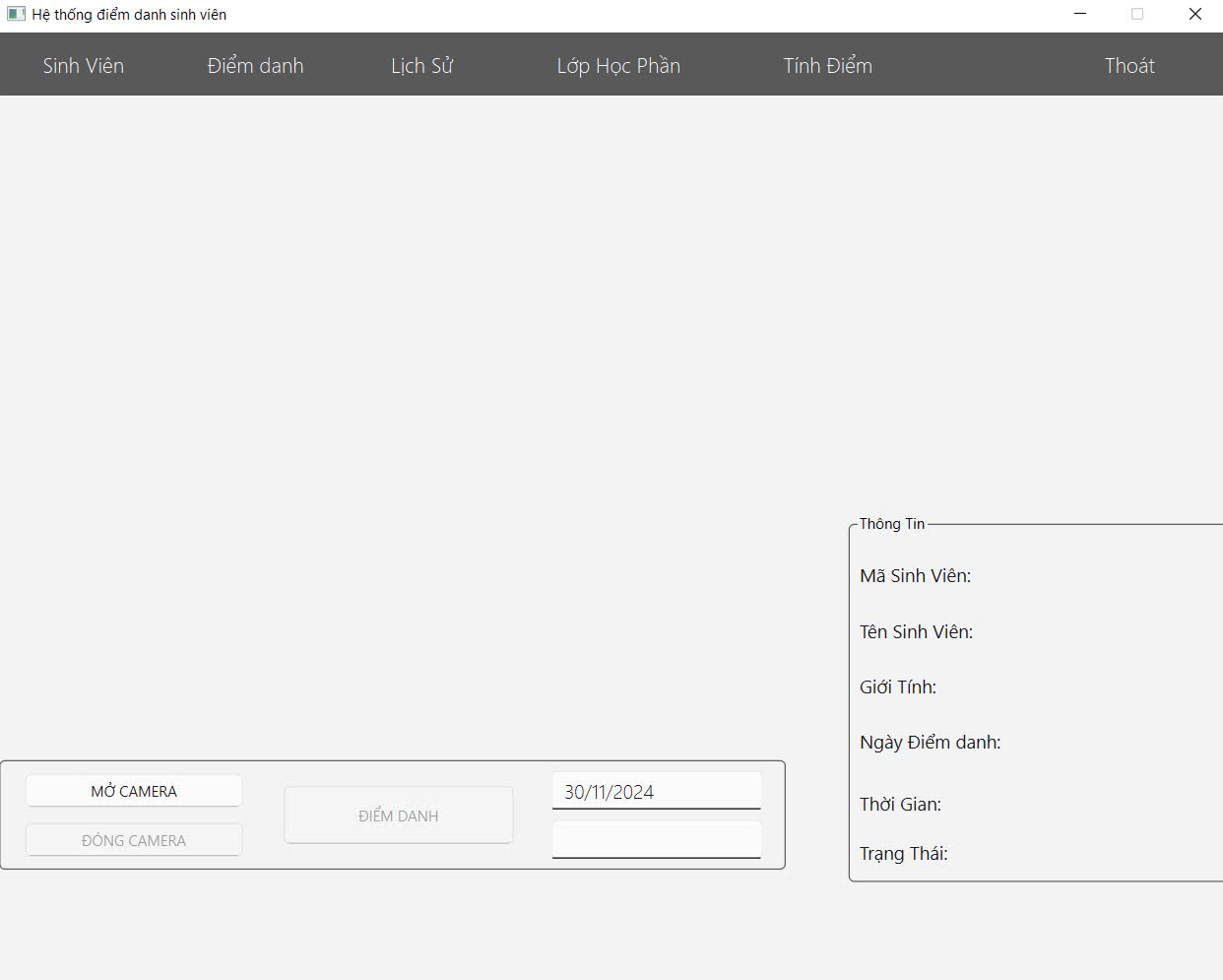
Xuất lịch sử điểm danh:xuất file Excel lịch sử điểm danh của sinh viên để theo dõi và lưu trữ.

### **3.2.3 Giao Diện Sản Phẩm**

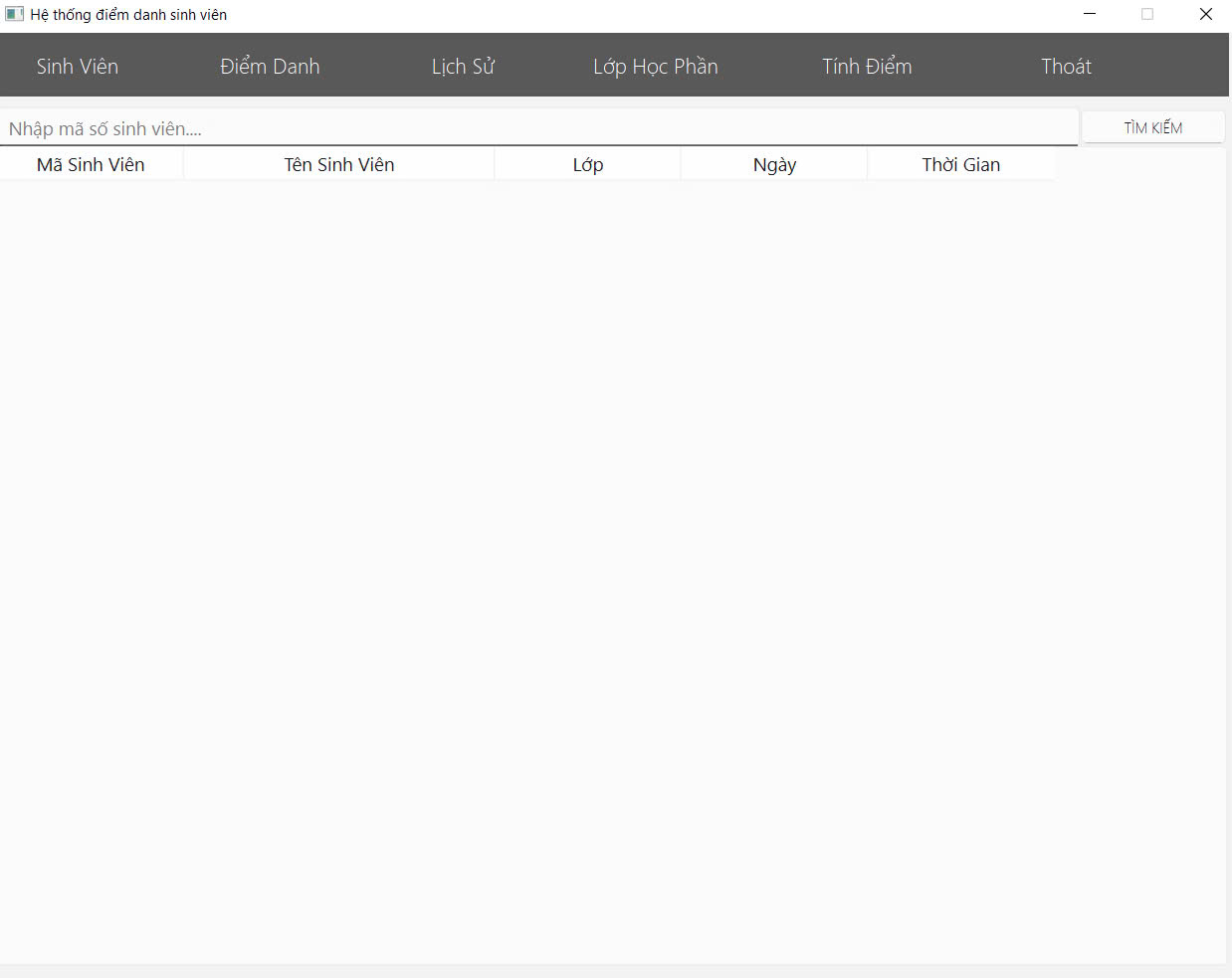
Dưới dây là giao diện sản phẩm của hệ thống nhận diện sinh viên



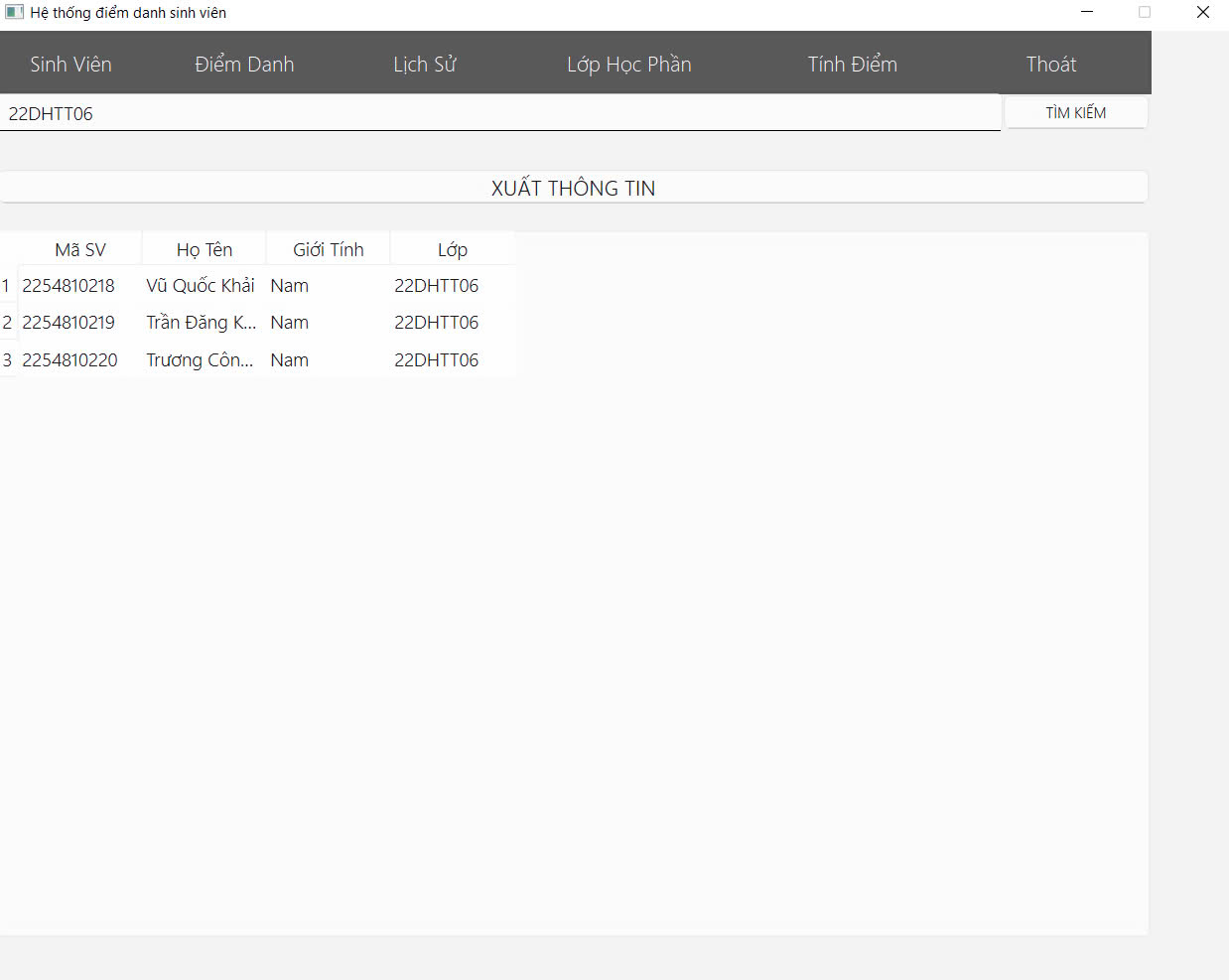
Hình 3.13. Giao diện của tính năng sinh viên



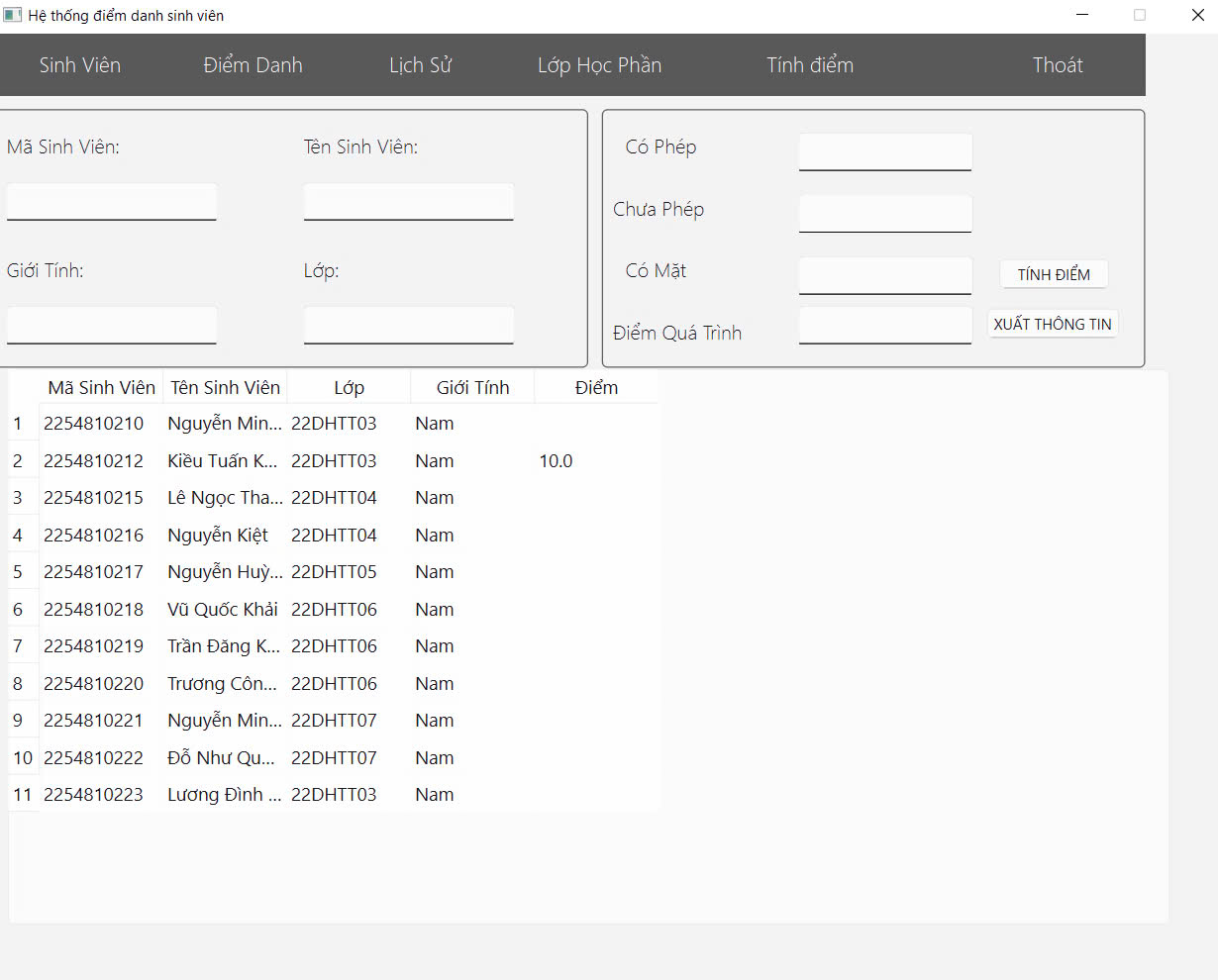
Hình 3.14. Giao diện của tính năng điểm danh



Hình 3.15. Giao diện của tính năng lịch sử điểm danh



Hình 3.16. Giao diện của tính năng chức năng lớp học phần



Hình 3.17. Giao diện của tính năng tính điểm chuyên cần

# **KẾT LUẬN**

Đề tài "Hệ thống điểm danh sinh viên bằng khuôn mặt" đã đạt được những kết quả đáng kể khi hoàn thiện các mục tiêu chính đề ra, bao gồm việc tự động hóa quy trình điểm danh, giảm thiểu sai sót và nâng cao hiệu quả trong quản lý lớp học. Sản phẩm được xây dựng với khả năng nhận diện khuôn mặt chính xác trong điều kiện ánh sáng ổn định, lưu trữ và quản lý dữ liệu điểm danh một cách an toàn trên cơ sở dữ liệu MySQL. Với giao diện thân thiện được phát triển bằng PyQt6 và ứng dụng các công nghệ hiện đại như OpenCV cùng Haar Cascade Classifier, hệ thống không chỉ hoạt động nhanh chóng mà còn dễ dàng sử dụng đối với người dùng. Tuy nhiên, trong suốt quá trình thực hiện, nhóm nhận thấy vẫn còn những hạn chế nhất định, chẳng hạn như khả năng nhận diện chưa tối ưu trong môi trường ánh sáng yếu, gặp khó khăn khi sinh viên đeo khẩu trang hoặc khi số lượng lớn sinh viên được điểm danh đồng thời. Bên cạnh đó, hệ thống chưa hỗ trợ các chức năng phân tích nâng cao như dự đoán xu hướng điểm danh hoặc thống kê chi tiết. Qua quá trình thực hiện đề tài, nhóm đã rút ra nhiều bài học quan trọng, đặc biệt là việc lập kế hoạch chi tiết hơn và thực hiện kiểm thử toàn diện để nâng cao chất lượng sản phẩm.

Trong tương lai, nhóm định hướng phát triển hệ thống theo hai khía cạnh chính: hoàn thiện các chức năng hiện tại và mở rộng tính năng mới. Về mặt hoàn thiện, nhóm sẽ nghiên cứu tích hợp các thuật toán nhận diện tiên tiến hơn như LBPH hoặc học sâu (deep learning) để nâng cao độ chính xác và cải thiện tốc độ xử lý, đảm bảo hiệu quả ngay cả khi xử lý số lượng lớn sinh viên. Đồng thời, giao diện người dùng sẽ được cải tiến với các tính năng trực quan hơn, chẳng hạn như biểu đồ thống kê dữ liệu hoặc hiển thị thông tin sinh viên theo thời gian thực. Về mở rộng, hệ thống sẽ được phát triển thêm một ứng dụng di động để giảng viên có thể quản lý và giám sát điểm danh từ xa, tích hợp công nghệ IoT để liên kết với các thiết bị thông minh, hoặc xây dựng các tính năng phân tích và dự đoán dữ liệu điểm danh dựa trên học máy, hỗ trợ nhà trường trong việc quản lý và đánh giá sinh viên. Bên cạnh đó, việc thử nghiệm hệ thống trong các bối cảnh khác như doanh nghiệp hoặc sự kiện lớn cũng là một hướng đi tiềm năng, đồng thời nâng cao tính bảo mật dữ liệu để đáp ứng các yêu cầu bảo vệ thông tin cá nhân trong môi trường thực tế. Những định hướng này không chỉ giúp cải thiện chất lượng sản phẩm mà còn mở ra cơ hội áp dụng công nghệ vào nhiều lĩnh vực khác, đóng góp tích cực cho sự phát triển của ngành công nghệ thông tin và giáo dục.

# **DANH MỤC TÀI LIỆU THAM KHẢO**

|  |  |
| --- | --- |
| [1] | R. Szeliski, Computer Vision: Algorithms and Applications, Springer, 2010. |
| [2] | K. Fukunaga, Introduction to Statistical Pattern Recognition, 2nd ed., Academic Press, 2013. |
| [3] | R. G. a. R. Woods, Digital Image Processing, 4th ed., Pearson, 2017. |
| [4] | S. S.-S. a. S. Ben-David, Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press, 2014. |
| [5] | Z. Zhang, Haar Cascade Algorithm in Real-Time Systems, Advances in Machine Learning and Vision, Springer, 2018. |
| [6] | G. B. a. A. Kaehler, Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library, 1st ed., O'Reilly Media, 2017. |
| [7] | M. D. Dhome, Face Detection using Haar Cascades in Embedded Systems, Embedded Vision Techniques: IEEE Press, 2021. |
| [8] | A. R. a. P. Das, Smart Attendance Monitoring System: A Haar-Cascade Approach, Springer, 2020. |
| [9] | C. Bishop, Pattern Recognition and Machine Learning, 2006. |
| [10] | M. T. a. A. Pentland, Face Recognition in Artificial Intelligence, Cambridge: MIT Press, 1991. |
| [11] | P. F. a. A. R. K. Jain, Handbook of Biometrics, Springer, 2007. |
| [12] | Z. Sun, Face Detection and Recognition Systems, Academic Press, 2012. |
| [13] | J. Canny, Biometric Attendance Systems: Concepts and Challenges, Wiley, 2015. |
| [14] | R. T. a. J. F. T. Hastie, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, 2009. |
| [15] | L. V. a. M. Jones, Rapid Object Detection using a Boosted Cascade of Simple Features, Proc. of IEEE CVPR, 2001. |