



TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO
PBL4 – DỰ ÁN HỆ THỐNG THÔNG MINH

HỆ THỐNG ĐIỂM DANH BẰNG KHUÔN MẶT

Real time Multi-Face recognition system



Giảng viên hướng dẫn: TS. Phạm Công Thắng

Họ và tên	Lớp sinh hoạt	Lớp học phần
Trần Lê Duy	22T_Nhat1	22.16A
Lê Văn Tuấn Lộc	22T_Nhat1	22.16A
Trần Thị Thanh Phương	22T_Nhat1	22.16A
Trần Minh Quang	22T_Nhat1	22.16A

ĐÀ NẴNG, 12/2024

TÓM TẮT ĐỒ ÁN

Trước tiên nhóm xin phép gửi lời cảm ơn tới TS. Phạm Công Thắng đã tận tình hỗ trợ, theo sát tiến độ của nhóm trong quá trình nhóm hoàn thành dự án này.

Hiện nay, việc quản lý điểm danh và theo dõi học sinh tại các trung tâm ngoại ngữ còn mất nhiều thời gian và dễ xảy ra sai sót, gây khó khăn trong việc nâng cao hiệu quả quản lý và đảm bảo tính minh bạch. Để giải quyết vấn đề này, nhóm đã xây dựng hệ thống nhận diện khuôn mặt thông minh sử dụng Raspberry Pi, camera tích hợp, và các thuật toán trí tuệ nhân tạo, kết hợp công nghệ IoT. Hệ thống hoạt động bằng cách nhận diện khuôn mặt học sinh, phân tích dữ liệu và gửi thông tin nhận dạng lên Server để hiển thị trên trang web quản lý của trung tâm.

Kết quả đạt được là hệ thống đã tự động hóa quy trình điểm danh, giúp giảm thiểu sai sót, tiết kiệm thời gian và công sức của nhân viên. Đồng thời, hệ thống hỗ trợ quản lý thông tin học sinh một cách chính xác và nhanh chóng, tạo sự thuận tiện trong việc theo dõi và báo cáo tình hình học tập của học sinh đến phụ huynh. Qua đó, hệ thống không chỉ tối ưu hóa công tác quản lý mà còn nâng cao trải nghiệm học tập tại các trung tâm ngoại ngữ, góp phần hiện đại hóa giáo dục trong thời đại số hóa.

BẢNG PHÂN CÔNG NHIỆM VỤ

Họ và tên	Nhiệm vụ	Tiến độ
Trần Lê Duy	<ul style="list-style-type: none">- Cấu hình Raspberry Pi- Dán nhãn dữ liệu	Đã hoàn thành
Lê Văn Tuấn Lộc	<ul style="list-style-type: none">- Cấu hình Raspberry Pi- Xây dựng model AI- Tạo server Websocket	Đã hoàn thành
Trần Thị Thanh Phương	<ul style="list-style-type: none">- Thiết kế, xây dựng giao diện- Xử lý back-end- Tạo front-end Websocket	Đã hoàn thành
Trần Minh Quang	<ul style="list-style-type: none">- Thiết kế, đồ dữ liệu database- Xử lý back-end- Dán nhãn dữ liệu	Đã hoàn thành

MỤC LỤC

1. Tổng quan đề tài	6
1.1. Đặt vấn đề	6
1.2. Mục tiêu đề tài	6
2. Giải pháp	7
2.1. Tổng quan	7
2.1.1. Các vấn đề và giải pháp đặt ra	7
2.1.2. Sơ đồ hoạt động	8
2.2. Thuật toán	9
2.2.1. Giới thiệu	9
2.2.2. Thuật toán huấn luyện mô hình (model_training.py)	9
2.2.3. Thuật toán chụp ảnh và lưu trữ (image_capture.py)	10
2.2.4. Thuật toán nhận diện khuôn mặt (facial_recognition.py)	10
2.3. Hệ thống	11
2.3.1. Chức năng	11
2.3.2. Sơ đồ usecase	11
2.3.3. Sơ đồ cơ sở dữ liệu	12
2.4. Phần cứng và truyền thông	12
2.4.1. Tổng quan hệ thống	12
2.4.2. Linh kiện và tham số kỹ thuật	13
2.5. Phần mềm	13
2.5.1. Nhận diện khuôn mặt trên Raspberry	13
2.5.2. Xử lý dữ liệu và điểm danh	17
2.5.3. Giao diện người dùng	19
3. Kết quả	20
3.1. Thu thập và xử lý dữ liệu	20
3.1.1. Thu thập dữ liệu	20
3.1.2. Xử lý dữ liệu	20
3.1.3. Phân chia tập dữ liệu	20
3.2. Công cụ và framework	21

3.2.1. Môi trường huấn luyện	21
3.2.2. Framework	21
3.3. Kết quả trên tập huấn luyện	22
3.4. Quy trình thực nghiệm và kiểm thử	22
3.5. Kết quả trên tập kiểm thử	23
3.5.1. Độ chính xác dự đoán (Precision)	23
3.5.2. Độ nhạy (Recall)	24
3.5.3. F1-Score	24
3.5.4. Độ chính xác tổng thể (Accuracy)	24
3.5.5. Confusion Matrix	25
3.6. Giao diện web	25
4. Kết luận	31
5. Danh mục tài liệu tham khảo	32

DANH MỤC HÌNH VẼ

Hình 1. Sơ đồ tổng quan hệ thống	8
Hình 2. Sơ đồ usecase.....	11
Hình 3. Sơ đồ cơ sở dữ liệu	12
Hình 4. Sequence Diagram: Huấn luyện nhận diện khuôn mặt	14
Hình 5. Sequence Diagram: Truyền dữ liệu Camera qua WebSocket	15
Hình 6. Sequence Diagram: Điểm danh	16
Hình 7. Sequence Diagram: Chụp và thêm khuôn mặt	17
Hình 8. Biểu đồ biểu hiện độ chính xác và mất mát qua các giai đoạn huấn luyện.....	22
Hình 9. Kết quả trên tập kiểm thử	23
Hình 10. Giao diện đăng nhập.....	25
Hình 11. Giao diện trang chủ	26
Hình 12. Giao diện tổng quan lớp học.....	26
Hình 13. Giao diện học viên trong một lớp học	27
Hình 14. Giao diện xem thông tin học viên.....	27
Hình 15. Giao diện sửa thông tin học viên.....	28
Hình 16. Giao diện thêm học viên mới	28
Hình 17. Giao diện chụp ảnh khi thêm học viên	29
Hình 18. Giao diện lịch học trong ngày	29
Hình 19. Giao diện điểm danh.....	30
Hình 20. Email điểm danh.....	30
Hình 21. Email vắng mặt.....	31

1. Tổng quan đề tài

1.1. Đặt vấn đề

Trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ tại thành phố Đà Nẵng, việc ứng dụng công nghệ hiện đại vào các lĩnh vực, đặc biệt là giáo dục, đã trở thành xu hướng tất yếu. Tuy nhiên, tại nhiều trung tâm ngoại ngữ, quy trình quản lý học sinh, đặc biệt là điểm danh và theo dõi tình hình học tập, vẫn còn phụ thuộc vào các phương pháp thủ công hoặc truyền thống. Những phương pháp này không chỉ tốn thời gian mà còn dễ gây ra sai sót, ảnh hưởng đến tính chính xác và hiệu quả của công tác quản lý.

Ngoài ra, sự kết nối và minh bạch trong việc thông báo tình hình học tập của học sinh đến phụ huynh cũng chưa được đảm bảo. Điều này có thể làm giảm chất lượng trải nghiệm của phụ huynh và học viên, đồng thời gây khó khăn cho trung tâm trong việc xây dựng uy tín và nâng cao chất lượng dịch vụ.

Trong bối cảnh đó, việc phát triển một hệ thống tự động hóa thông minh, kết hợp giữa trí tuệ nhân tạo và Internet vạn vật (IoT), để giải quyết những bất cập hiện tại là một yêu cầu cấp thiết. Hệ thống không chỉ cần đảm bảo tính chính xác và tiết kiệm thời gian mà còn phải dễ sử dụng, đáp ứng nhu cầu theo dõi, quản lý minh bạch và kịp thời.

Dựa trên thực tiễn và nhu cầu này, nhóm chúng tôi đã quyết định triển khai đề tài xây dựng hệ thống nhận diện khuôn mặt thông minh. Hệ thống ứng dụng Raspberry Pi, camera tích hợp, cùng các thuật toán xử lý ảnh và học máy, nhằm tự động hóa quy trình điểm danh, quản lý thông tin học sinh và nâng cao hiệu quả hoạt động tại các trung tâm ngoại ngữ.

1.2. Mục tiêu đề tài

Xây dựng hệ thống nhận diện khuôn mặt thông minh, ứng dụng công nghệ trí tuệ nhân tạo (AI) và Internet vạn vật (IoT), nhằm tự động hóa quy trình điểm danh và quản lý học sinh tại các trung tâm ngoại ngữ, từ đó nâng cao hiệu quả quản lý và trải nghiệm học tập.

Tự động hóa quy trình điểm danh: Hệ thống sử dụng công nghệ nhận diện khuôn mặt chính xác và nhanh chóng, thay thế các phương pháp thủ công truyền thống. Điều này giúp tiết kiệm thời gian và đảm bảo tính chính xác trong việc xác định danh tính học sinh.

Quản lý thông tin học sinh: Thông tin nhận diện được lưu trữ và xử lý trên server, đồng thời tích hợp với giao diện trang web quản lý của trung tâm. Dữ liệu được hiển thị trực quan, dễ sử dụng, hỗ trợ việc tra cứu và quản lý thông tin nhanh chóng.

Tăng tính minh bạch: Hệ thống cung cấp thông tin chi tiết về tình hình học tập và chuyên cần của học sinh, giúp phụ huynh theo dõi kịp thời. Điều này góp phần tăng cường niềm tin và sự hài lòng của phụ huynh đối với trung tâm ngoại ngữ.

Tiết kiệm thời gian và nguồn lực: Bằng cách tự động hóa quy trình, hệ thống giúp giảm bớt khối lượng công việc thủ công, tối ưu hóa nguồn lực và nâng cao hiệu quả vận hành của trung tâm.

Phát triển giải pháp công nghệ hiện đại: Ứng dụng các thuật toán trí tuệ nhân tạo và học máy (Machine Learning) nhằm đảm bảo độ chính xác cao trong quá trình nhận diện khuôn mặt, đồng thời đưa ra các cải tiến phù hợp với nhu cầu thực tế.

Thích nghi với xu hướng chuyển đổi số: Hệ thống đáp ứng yêu cầu phát triển trong thời đại số hóa, góp phần hiện đại hóa công tác quản lý giáo dục tại các trung tâm, phù hợp với xu hướng xây dựng thành phố thông minh và chuyển đổi số.

2. Giải pháp

2.1. Tổng quan

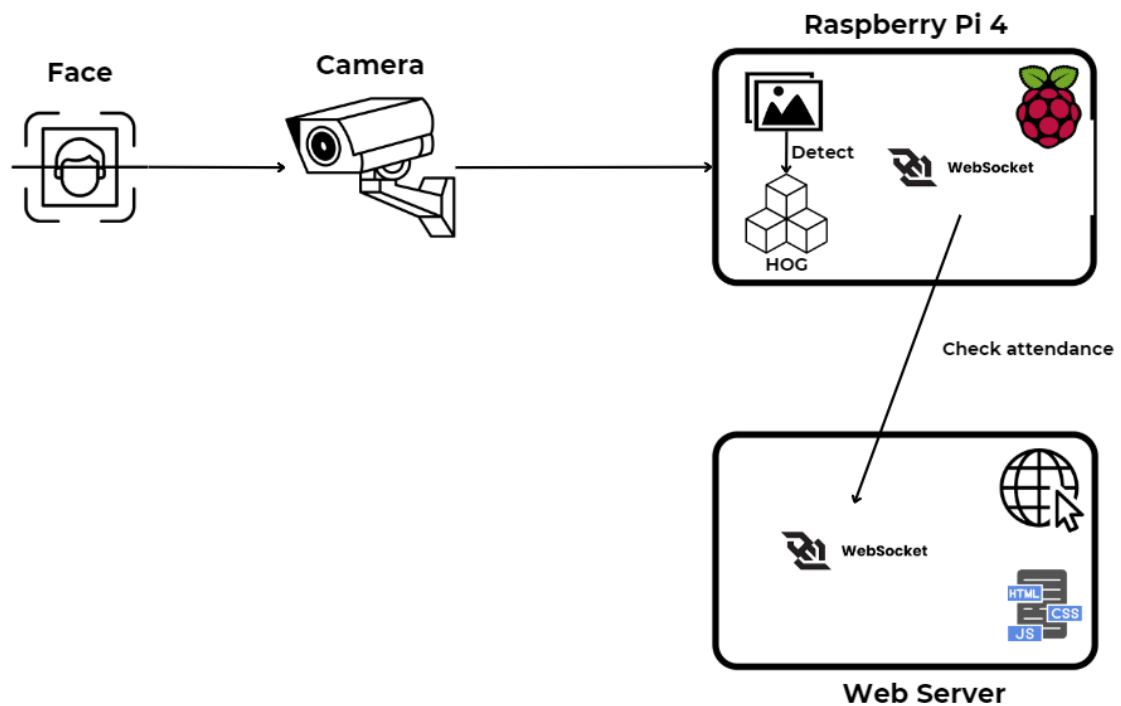
2.1.1. Các vấn đề và giải pháp đặt ra

Bảng 1. Bảng giải pháp đề xuất cho các vấn đề

Vấn đề	Đề xuất giải pháp
Phần cứng	<ul style="list-style-type: none"> - Raspberry Pi 4 - Camera
Nhận diện khuôn mặt	<ul style="list-style-type: none"> - Tối ưu hóa hệ điều hành Raspberry Pi OS - Huấn luyện và tinh chỉnh mô hình - Cải tiến thuật toán nhận diện
Ứng dụng Web	<ul style="list-style-type: none"> - Xây dựng giao diện web dễ sử dụng - Tích hợp với server và cơ sở dữ liệu - Đảm bảo bảo mật và quyền truy cập

2.1.2. Sơ đồ hoạt động

Sơ đồ hoạt động tổng quan của hệ thống được thể hiện qua *hình 1*:



Hình 1. Sơ đồ tổng quan hệ thống

2.2. Thuật toán

2.2.1. Giới thiệu

Dự án nhận diện khuôn mặt trong bài toán này được chia thành ba phần chính: Huấn luyện mô hình nhận diện khuôn mặt, Chụp ảnh và lưu trữ ảnh cho người dùng, và Nhận diện khuôn mặt từ camera. Các thuật toán sử dụng trong từng phần nhằm mục đích nhận diện và xác định các khuôn mặt trong các ảnh chụp, từ đó lưu trữ các thông tin nhận diện cho mục đích xác thực sau này.

2.2.2. Thuật toán huấn luyện mô hình (`model_training.py`)

Quá trình huấn luyện mô hình nhận diện khuôn mặt được thực hiện qua các bước:

- Tiền xử lý ảnh:
 - + Sử dụng thư viện `face_recognition` để nhận diện và mã hóa khuôn mặt. Các ảnh đầu vào được chia thành 10 giai đoạn (`num_stages`), mỗi giai đoạn xử lý một tập hợp ảnh con từ thư mục `dataset`.
 - + HOG (Histogram of Oriented Gradients): Được sử dụng trong thư viện `face_recognition` để phát hiện vị trí của khuôn mặt trong ảnh. Đây là một kỹ thuật phổ biến để nhận diện các đặc điểm khuôn mặt trong ảnh.
 - + Mỗi ảnh sau khi được đọc từ thư mục sẽ được chuyển đổi từ BGR sang RGB (vì `face_recognition` yêu cầu ảnh RGB).
 - + Sử dụng phương pháp `face_locations` để phát hiện vị trí khuôn mặt trong ảnh và `face_encodings` để tạo mã hóa khuôn mặt.
- Tính toán Accuracy và Loss: Đối với mỗi giai đoạn, thuật toán tính toán tỷ lệ chính xác (`accuracy`) và loss giả lập (`difference between 100% and accuracy`). Kết quả được lưu lại để vẽ đồ thị.
- Lưu trữ kết quả:
 - + Sau khi hoàn thành quá trình mã hóa khuôn mặt, các mã hóa khuôn mặt và tên được lưu vào tệp `encodings.pickle` để sử dụng trong các bước nhận diện sau này.
 - + Đồ thị về `accuracy` và `loss` được vẽ để đánh giá sự tiến bộ qua các giai đoạn.

2.2.3. Thuật toán chụp ảnh và lưu trữ (image_capture.py)

Thuật toán chụp ảnh sử dụng camera của Raspberry Pi để thu thập ảnh của người dùng:

- Tạo thư mục lưu trữ ảnh:
 - + Một ID mới cho mỗi người sẽ được tạo tự động bằng cách lấy ID cao nhất hiện có trong thư mục dataset và tăng lên một.
 - + Một thư mục mới được tạo ra cho mỗi người dùng với tên là ID đó.
- Chụp ảnh:
 - + Sử dụng camera Raspberry Pi (thư viện picamera2) để chụp ảnh.
 - + Người dùng có thể nhấn phím 'c' để chụp 20 ảnh liên tiếp. Các ảnh này sẽ được lưu với tên file chứa dấu thời gian để đảm bảo mỗi ảnh có tên duy nhất.
- Cấu hình camera: Cấu hình camera để quay với độ phân giải 640x640, sau đó lưu ảnh vào thư mục đã tạo.

2.2.4. Thuật toán nhận diện khuôn mặt (facial_recognition.py)

Thuật toán nhận diện khuôn mặt sử dụng ảnh thu được từ camera và so sánh chúng với các mã hóa khuôn mặt đã được huấn luyện trước:

- Tải mã hóa khuôn mặt: Các mã hóa khuôn mặt từ tệp encodings.pickle được tải và sử dụng để nhận diện khuôn mặt trong các khung hình.
- Quá trình nhận diện khuôn mặt:
 - + Mỗi khung hình từ camera sẽ được thu nhỏ lại (scaling) để tăng tốc quá trình xử lý.
 - + Sử dụng face_recognition.face_locations để phát hiện vị trí khuôn mặt và face_recognition.face_encodings để mã hóa khuôn mặt trong từng khung hình.
 - + Các mã hóa khuôn mặt phát hiện được so sánh với các mã hóa đã lưu trước đó bằng cách sử dụng phương thức compare_faces và tính toán độ tương đồng giữa chúng.

- Đánh giá độ chính xác:
 - + Độ chính xác giữa khuôn mặt phát hiện và các khuôn mặt đã biết được tính toán dựa trên khoảng cách giữa các mã hóa khuôn mặt. Nếu độ chính xác thấp hơn 60%, khuôn mặt được gán là "Unknown".
 - + Ngược lại, nếu khuôn mặt quét được trùng với khuôn mặt đã có trong dataset, thì hệ thống sẽ hiện ID của người đó lên màn hình.

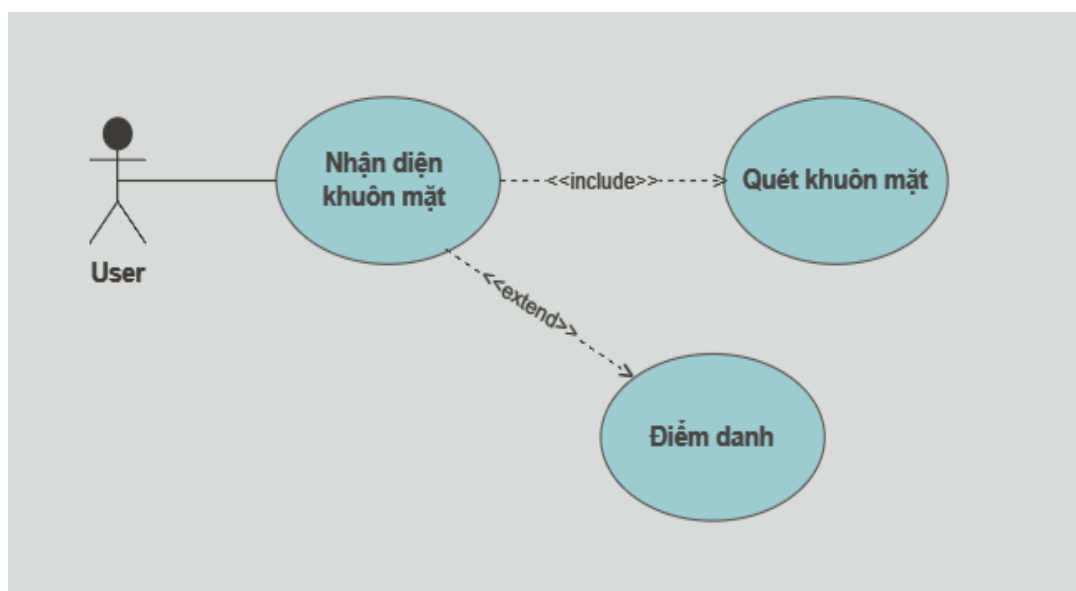
2.3. Hệ thống

2.3.1. Chức năng

- Nhận diện và xác thực khuôn mặt.
- Điểm danh.
- Lưu trữ và quản lý dữ liệu khuôn mặt của học viên.
- Gửi thông báo tự động qua email cho học viên vắng mặt.

2.3.2. Sơ đồ usecase

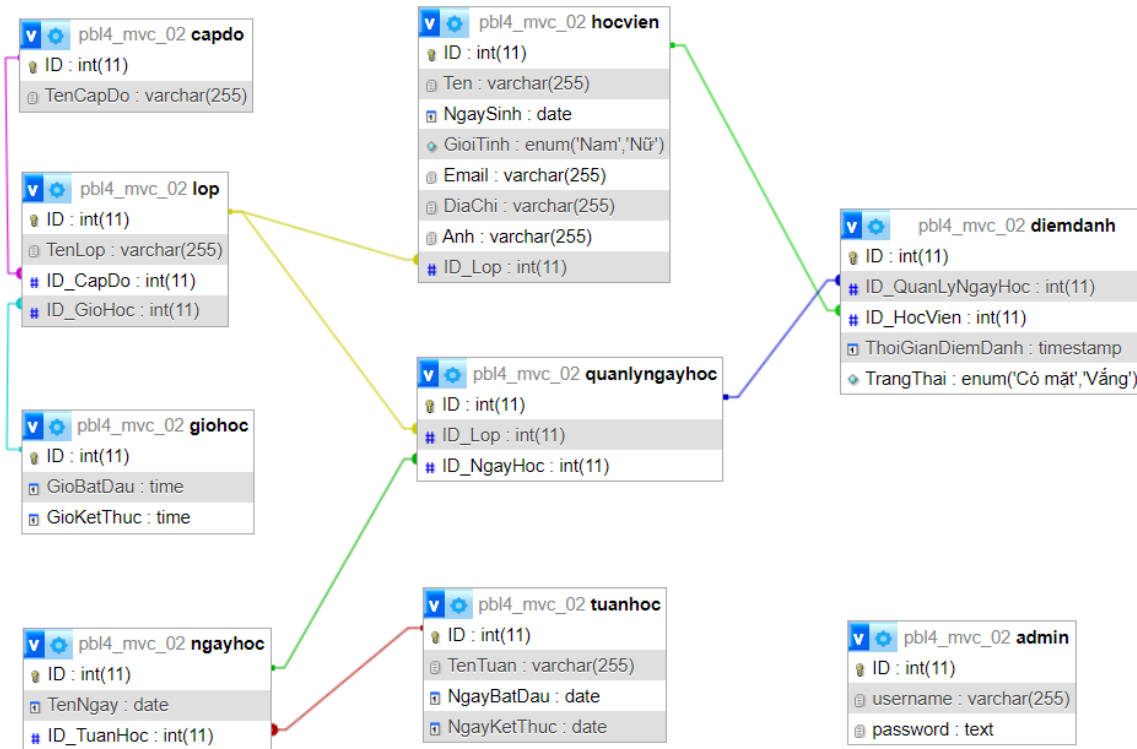
Sơ đồ usecase được thể hiện qua hình 2:



Hình 2. Sơ đồ usecase

2.3.3. Sơ đồ cơ sở dữ liệu

Sơ đồ cơ sở dữ liệu được thể hiện qua hình 3:



Hình 3. Sơ đồ cơ sở dữ liệu

2.4. Phần cứng và truyền thông

2.4.1. Tổng quan hệ thống

Hệ thống nhận diện khuôn mặt tự động dựa trên nền tảng Raspberry Pi bao gồm:

- Camera:
 - + Chụp ảnh khuôn mặt học viên.
 - + Truyền hình ảnh trực tiếp đến Raspberry Pi để xử lý.
- Raspberry Pi 4:
 - + Xử lý dữ liệu hình ảnh từ camera.
 - + Chạy các thuật toán nhận diện khuôn mặt.
 - + Gửi dữ liệu điểm danh đến webserver thông qua mạng LAN/WiFi.

2.4.2. Linh kiện và tham số kỹ thuật

Bảng 2. Danh sách linh kiện và tham số kỹ thuật

STT	Tên linh kiện	Tham số kỹ thuật	Nguyên tắc hoạt động
1	Raspberry Pi 4	4GB RAM, 4vCPU	Là một máy tính nhúng, có thể cài đặt hệ điều hành và có giao diện tương tác trực tiếp qua bàn phím, chuột, màn hình.
2	Raspberry Pi Camera Module V2 8MP	Độ phân giải 1080P	Thu nhận ánh sáng, chuyển đổi thành tín hiệu số, và truyền dữ liệu hình ảnh qua giao tiếp MIPI CSI-2 đến Raspberry Pi.
3	Thẻ nhớ SD	32GB	Sử dụng công nghệ bộ nhớ flash NAND để lưu trữ dữ liệu dưới dạng các ô nhớ có thể xóa và ghi lại nhiều lần.

2.5. Phần mềm

2.5.1. Nhận diện khuôn mặt trên Raspberry

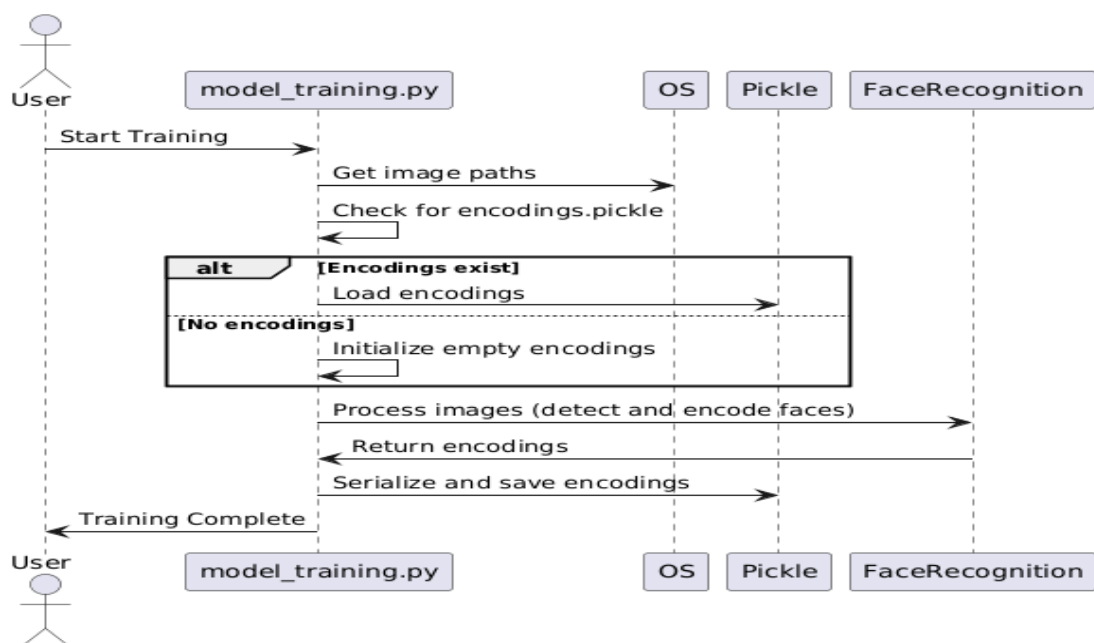
a. Tổng quan

- Kết nối và đọc video từ camera với Raspberry Pi: Sử dụng thư viện OpenCV, Raspberry Pi sẽ kết nối với camera để nhận dạng khuôn mặt qua các video trực tiếp. Việc xử lý frame video sẽ được thực hiện ngay trên Raspberry Pi, với các thông số và thao tác phù hợp.
- Xử lý nhận diện khuôn mặt trên Raspberry Pi: Mô hình nhận diện khuôn mặt sử dụng định dạng Pytorch Model, giúp tối ưu hiệu suất khi thực hiện nhận diện trên máy tính nhúng. Mô hình sẽ trả về các thông số bounding-box và độ tin cậy (confidence) của từng dự đoán khuôn mặt. Dữ liệu này sau đó được xử lý và gửi về server.

- Gửi yêu cầu lên Server: Thông qua module requests của Python, các dữ liệu liên quan đến khuôn mặt nhận diện sẽ được gửi qua giao thức HTTP đến server, bao gồm thông tin về ID khuôn mặt và tỷ lệ chính xác.

b. Sequence Diagram 1: Huấn luyện nhận diện khuôn mặt

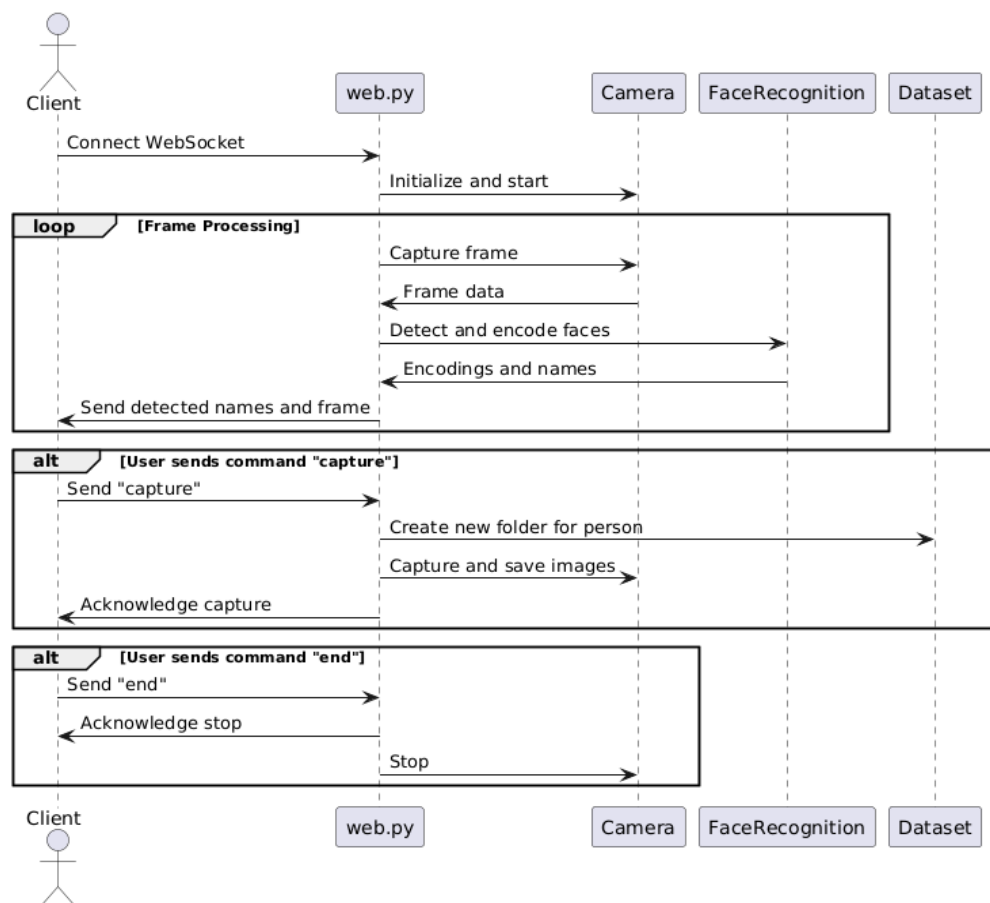
- Đối tượng:
 - + User: Người dùng khởi tạo quá trình training ảnh khuôn mặt.
 - + Model_training.py: Mã Python đảm nhiệm quá trình training và xử lý ảnh.
 - + OS: Hệ thống tệp tin, giúp lấy đường dẫn tới các ảnh.
 - + Pickle: Công cụ để lưu trữ và nạp dữ liệu (encoding) khuôn mặt đã xử lý.
 - + FaceRecognition: Thư viện hoặc module xử lý nhận diện khuôn mặt.
- Giải thích: Sơ đồ này minh họa quy trình khi người dùng khởi động quá trình mã hóa (encoding) ảnh khuôn mặt từ thư mục dataset. Nếu tệp encodings.pickle đã tồn tại, hệ thống nạp dữ liệu mã hóa từ đó. Nếu không, nó khởi tạo tệp mới và bắt đầu xử lý các ảnh. Các ảnh được xử lý bằng thư viện nhận diện khuôn mặt để phát hiện và mã hóa khuôn mặt, sau đó lưu kết quả vào tệp encodings.pickle.
- Quá trình huấn luyện nhận diện khuôn mặt được thể hiện qua hình 4:



Hình 4. Sequence Diagram: Huấn luyện nhận diện khuôn mặt

c. Sequence Diagram 2: Truyền dữ liệu Camera qua WebSocket

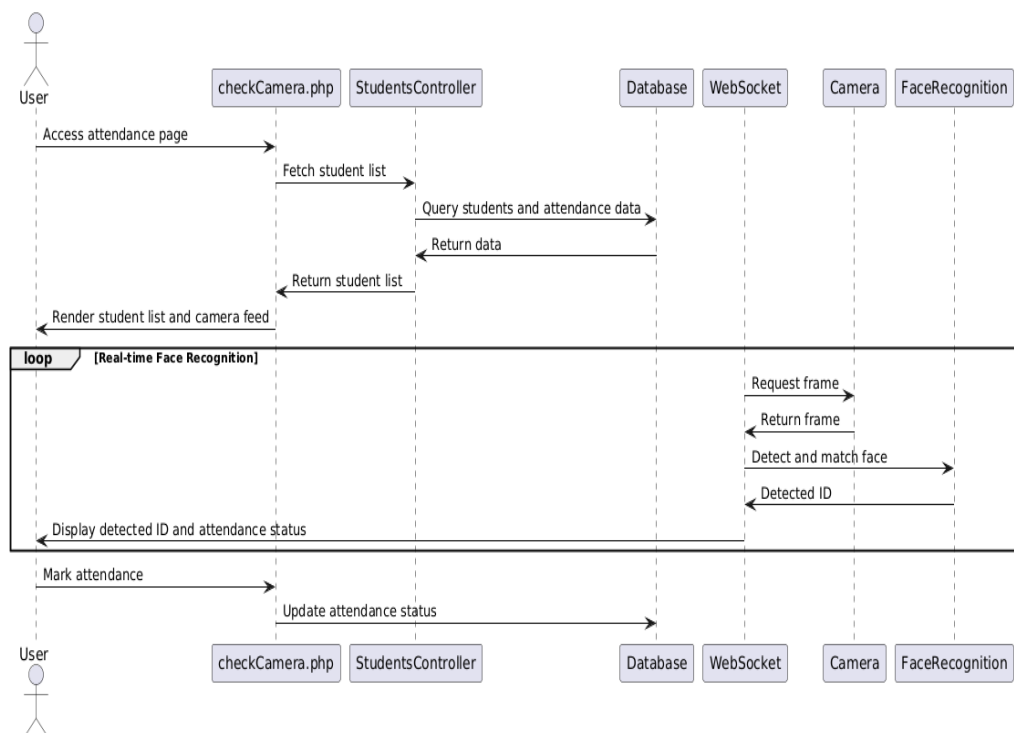
- Đối tượng:
 - + Client: Người dùng giao tiếp với server thông qua WebSocket.
 - + web.py: Mã Python chạy server WebSocket, xử lý luồng video từ camera.
 - + Camera: Thiết bị cung cấp luồng video.
 - + FaceRecognition: Thư viện xử lý nhận diện khuôn mặt.
 - + Dataset: Nơi lưu trữ dữ liệu ảnh khi có lệnh chụp từ người dùng.
- Giải thích: Sơ đồ này mô tả cách server WebSocket nhận luồng video từ camera và gửi dữ liệu nhận diện (tên và hình ảnh) về client. Trong vòng lặp xử lý, server nhận khung hình từ camera, phát hiện khuôn mặt và gửi kết quả nhận diện về client. Nếu người dùng gửi lệnh "capture", hệ thống sẽ tạo một thư mục mới và lưu các ảnh từ camera. Nếu gửi lệnh "end", server dừng camera và ngắt kết nối.
- Quá trình truyền dữ liệu Camera qua WebSocket được thể hiện qua *hình 5*:



Hình 5. Sequence Diagram: Truyền dữ liệu Camera qua WebSocket

d. Sequence Diagram 3: Điểm danh

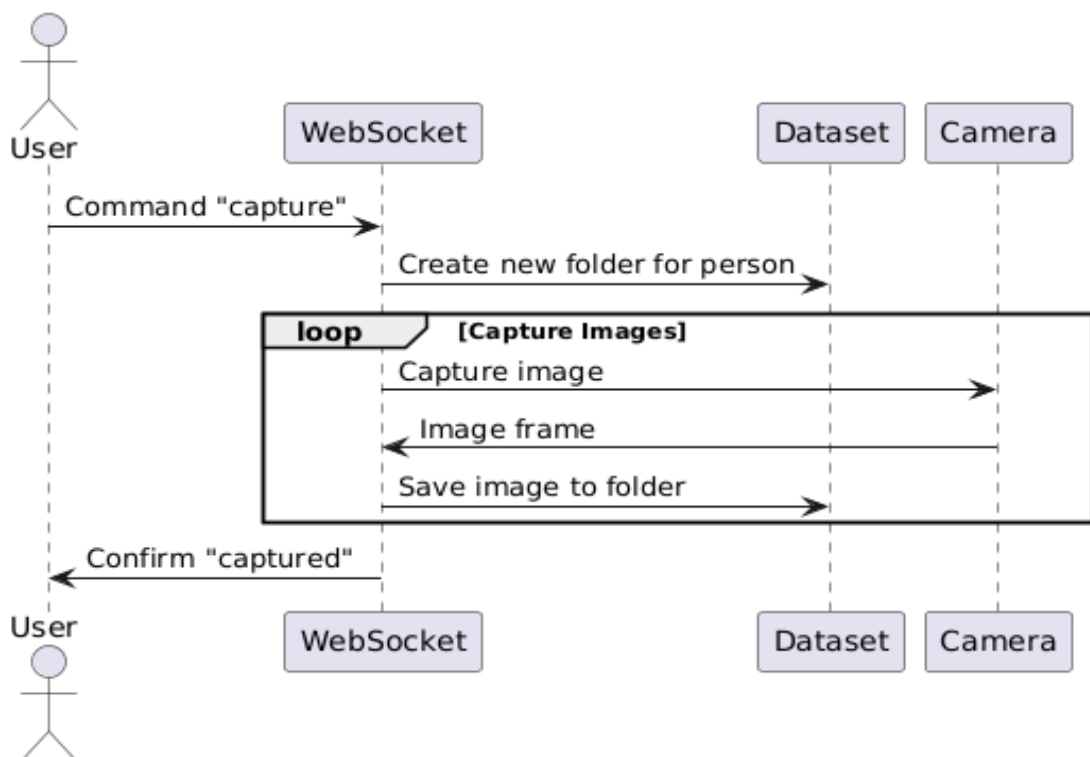
- Đối tượng:
 - + User: Người dùng truy cập vào hệ thống điểm danh học sinh.
 - + CheckCamera.php: Trang PHP xử lý hiển thị giao diện điểm danh và camera.
 - + StudentsController: Controller lấy dữ liệu danh sách học sinh và thông tin điểm danh.
 - + Database: Cơ sở dữ liệu chứa danh sách học sinh và trạng thái điểm danh.
 - + WebSocket: Thành phần xử lý giao tiếp giữa camera và trình duyệt.
 - + Camera: Thiết bị cung cấp luồng video.
 - + FaceRecognition: Thư viện xử lý nhận diện khuôn mặt.
- Giải thích: Sơ đồ này trình bày luồng xử lý khi người dùng mở trang điểm danh. Controller truy vấn dữ liệu từ database và trả về danh sách học sinh để hiển thị. Trong vòng lặp nhận diện, camera gửi khung hình tới server để nhận diện khuôn mặt và trả lại ID nhận diện cho người dùng. Người dùng có thể đánh dấu trạng thái điểm danh trong hệ thống.
- Quá trình điểm danh được thể hiện qua *hình 6*:



Hình 6. Sequence Diagram: Điểm danh

e. Sequence Diagram 4: Chụp và thêm khuôn mặt

- Đối tượng:
 - + User: Người dùng gửi lệnh chụp ảnh để thêm người mới.
 - + WebSocket: Server WebSocket xử lý lệnh và giao tiếp với camera.
 - + Dataset: Nơi lưu trữ các ảnh đã chụp.
 - + Camera: Thiết bị cung cấp khung hình ảnh.
- Giải thích: Sơ đồ mô tả quá trình thêm người mới vào hệ thống. Người dùng gửi lệnh "capture", server tạo một thư mục mới để lưu ảnh. Trong vòng lặp, camera chụp các khung hình, gửi ảnh về server để lưu. Khi hoàn tất, server xác nhận và thông báo lại cho người dùng.
- Quá trình chụp và thêm khuôn mặt được biểu hiện qua *hình 7*:



Hình 7. Sequence Diagram: Chụp và thêm khuôn mặt

2.5.2. Xử lý dữ liệu và điểm danh

- Thu thập dữ liệu khuôn mặt: Hệ thống nhận diện khuôn mặt bắt đầu bằng việc thu thập dữ liệu hình ảnh khuôn mặt của học sinh. Quy trình bao gồm:
 - + Chụp ảnh khuôn mặt: Sử dụng camera tích hợp với Raspberry Pi để chụp ảnh khuôn mặt học sinh từ nhiều góc độ.

- + Tiền xử lý hình ảnh:
 - Chuyển đổi ảnh màu sang ảnh xám (grayscale) để giảm kích thước và tăng hiệu quả xử lý.
 - Cân bằng ánh sáng và loại bỏ nhiễu hình ảnh để đảm bảo chất lượng đầu vào.
- Trích xuất đặc trưng khuôn mặt
 - + Sử dụng thư viện Face Recognition:
 - Ảnh đã qua tiền xử lý được đưa vào mô hình nhận diện khuôn mặt dựa trên Dlib.
 - Khuôn mặt được mã hóa thành một vector đặc trưng (embedding), đại diện cho các đặc điểm khuôn mặt duy nhất.
 - + Lưu trữ dữ liệu đặc trưng.
 - + Các vector đặc trưng khuôn mặt được lưu vào cơ sở dữ liệu (database) trên server cùng với thông tin cá nhân của học sinh (mã số, họ tên, lớp học,..).
- Nhận diện khuôn mặt thời gian thực:
 - + Phát hiện khuôn mặt trong khung hình:
 - Camera liên tục chụp và gửi hình ảnh đến Raspberry Pi.
 - OpenCV thực hiện phát hiện khuôn mặt trong khung hình và cắt ra vùng khuôn mặt.
- So sánh và nhận diện:
 - + Vector đặc trưng của khuôn mặt hiện tại được tạo ra và so sánh với các vector trong cơ sở dữ liệu.
 - + Sử dụng độ đo khoảng cách (chẳng hạn, Euclidean distance) để xác định mức độ tương đồng.
- Điểm danh và cập nhật dữ liệu: Ghi nhận trạng thái điểm danh, nếu khuôn mặt được nhận diện thành công và đạt độ tin cậy cao (>60%), hệ thống sẽ đánh dấu “✓”.
- Hiển thị trên giao diện web: Thông tin điểm danh được cập nhật thời gian thực và hiển thị trực quan trên trang web quản lý của trung tâm ngoại ngữ.

- Gửi thông tin qua email:
 - + Sau khi điểm danh, thông tin bao gồm tên học sinh, trạng thái điểm danh sẽ được gửi đến email của phụ huynh qua SMTP (Simple Mail Transfer Protocol).
 - + Nội dung email được thiết kế đơn giản, dễ hiểu với đầy đủ thông tin cần thiết để phụ huynh tiện theo dõi.

2.5.3. Giao diện người dùng

a. Giao diện

- Giao diện trang chủ: Thống kê số lượng học sinh, lớp học, lịch học của các lớp.
- Giao diện quản lý điểm danh: Hiển thị danh sách học sinh theo lớp với các cột, hàng: Mã học sinh, Tên học sinh, Trạng thái điểm danh (✓/✗).
- Giao diện thông tin học sinh: Hiển thị thông tin chi tiết: Họ tên, lớp học, ngày sinh, gmail...
- Frontend: Sử dụng HTML, CSS, JavaScript để thiết kế giao diện thân thiện và trực quan.

b. Tương tác với server

- Web Server:
 - + Được xây dựng bằng PHP, đảm bảo lưu trữ và xử lý thông tin điểm danh.
 - + Kết nối với cơ sở dữ liệu (MySQL) để lưu trữ dữ liệu học sinh.
- REST API:
 - + API được sử dụng để truyền thông tin giữa Raspberry Pi và server.
 - + Bao gồm các phương thức chính như:
 - POST: Gửi thông tin nhận diện lên server.
 - GET: Truy xuất thông tin học sinh hoặc báo cáo điểm danh

3. Kết quả

3.1. Thu thập và xử lý dữ liệu

3.1.1. Thu thập dữ liệu

Trong quá trình xây dựng hệ thống nhận diện khuôn mặt, việc thu thập dữ liệu đóng vai trò quan trọng nhằm đảm bảo độ chính xác và hiệu quả của hệ thống. Các bước thu thập dữ liệu được thực hiện như sau:

- Xác định yêu cầu dữ liệu: Dữ liệu cần thu thập bao gồm hình ảnh khuôn mặt học sinh từ nhiều góc độ khác nhau để đảm bảo khả năng nhận diện trong điều kiện thực tế.
- Thu thập hình ảnh: Sử dụng camera Raspberry Pi để chụp ảnh khuôn mặt học sinh trong điều kiện ánh sáng tiêu chuẩn. Yêu cầu học sinh thực hiện nhiều lần chụp từ các góc nhìn khác nhau để đảm bảo hệ thống nhận diện được cả trong trường hợp ánh sáng yếu hoặc khi có thay đổi góc nhìn.

3.1.2. Xử lý dữ liệu

Trong hệ thống, mỗi bức ảnh trong tập dữ liệu được lưu trữ trong một thư mục, và tên thư mục được sử dụng làm nhãn (label) cho đối tượng trong ảnh. Ví dụ, nếu tập dữ liệu có cấu trúc dataset/1/img1.jpg, tên thư mục 1 sẽ được dùng làm nhãn cho ảnh img1.jpg. Để thực hiện điều này, đường dẫn của ảnh được phân tích để trích xuất tên thư mục cha bằng cách tách chuỗi. Sau đó, ảnh được đọc và chuyển đổi từ không gian màu BGR (mặc định của OpenCV) sang RGB để tương thích với thư viện face_recognition.

3.1.3. Phân chia tập dữ liệu

- Có khoảng ~1100 hình ảnh đã được dán nhãn.
- Chia ngẫu nhiên dữ liệu hình ảnh theo tỉ lệ 80% cho tập train và 20% cho tập test.

3.2. Công cụ và framework

3.2.1. Môi trường huấn luyện

Môi trường huấn luyện cho mô hình nhận diện khuôn mặt sử dụng các công cụ và thư viện Python. Các thư viện chính được sử dụng bao gồm:

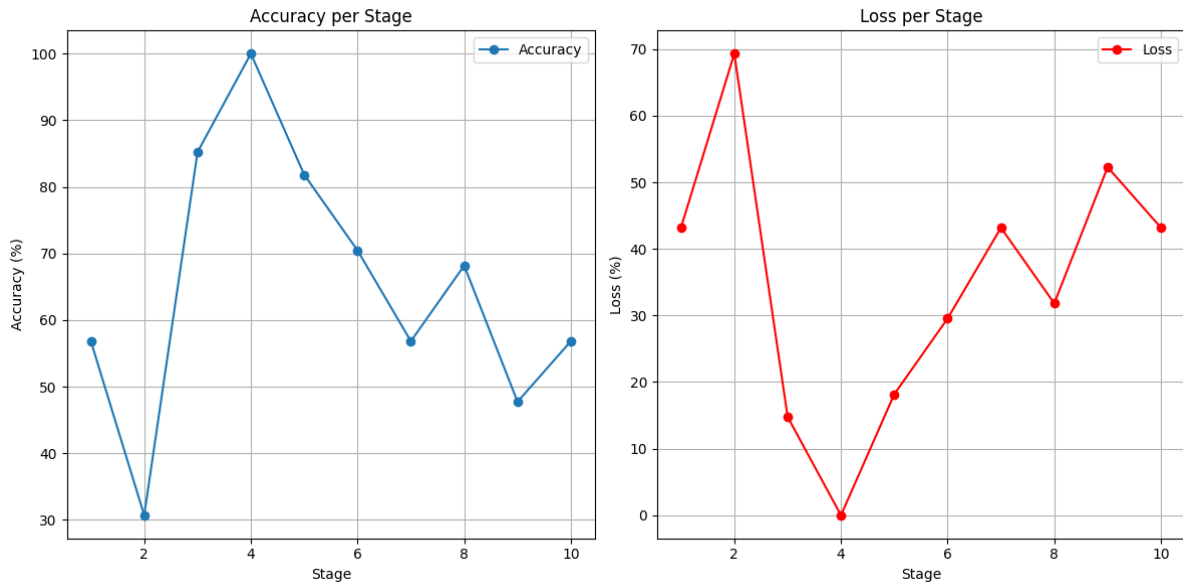
- OpenCV: Được sử dụng để xử lý hình ảnh và video, giúp lấy khung hình từ camera, hiển thị kết quả, và lưu trữ ảnh.
- Face Recognition: Thư viện này giúp phát hiện khuôn mặt trong ảnh, mã hóa các khuôn mặt và so sánh chúng để nhận diện người. Nó sử dụng các thuật toán học máy như HOG (Histogram of Oriented Gradients) và các mô hình học sâu như large model.
- Pickle: Dùng để lưu trữ và tái sử dụng các encoding khuôn mặt đã huấn luyện (pickle file) nhằm tiết kiệm thời gian trong các lần huấn luyện sau.

3.2.2. Framework

- Face Recognition: Là framework chính sử dụng để nhận diện khuôn mặt. Nó cung cấp các API để:
 - + Phát hiện khuôn mặt trong ảnh (face_locations).
 - + Mã hóa khuôn mặt (face_encodings).
 - + So sánh khuôn mặt để nhận diện người dùng bằng cách so sánh với các khuôn mặt đã được huấn luyện trước.
- OpenCV: Framework này giúp xử lý ảnh và video, bao gồm các chức năng như đọc ảnh, hiển thị, chụp ảnh từ camera và vẽ hình chữ nhật xung quanh các khuôn mặt phát hiện.
- Pickle: Framework này không chỉ giúp lưu trữ các mô hình (encodings) mà còn dễ dàng chia sẻ giữa các bước huấn luyện hoặc ứng dụng khác.

3.3. Kết quả trên tập huấn luyện

Độ chính xác theo từng giai đoạn (Accuracy per stage) và hàm mất mát theo từng giai đoạn (Loss per stage) được thể hiện qua *hình 8*:



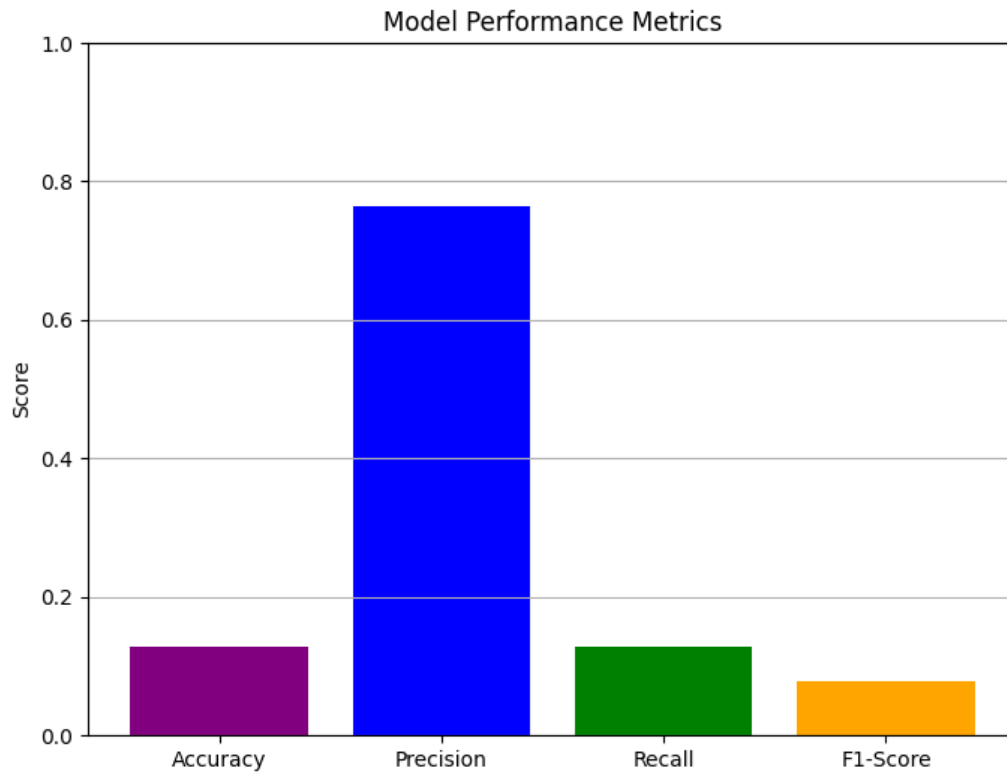
Hình 8. Biểu đồ biểu hiện độ chính xác và mất mát qua các giai đoạn huấn luyện

3.4. Quy trình thực nghiệm và kiểm thử

- Kết nối Raspberry với Camera để kiểm thử hiệu suất mô hình hoạt động.
- Thực hiện nhận diện khuôn mặt, kiểm tra độ tương thích và thực hiện điểm danh.

3.5. Kết quả trên tập kiểm thử

Kết quả trên tập kiểm thử được thể hiện qua *hình 9*:



Hình 9. Kết quả trên tập kiểm thử

3.5.1. Độ chính xác dự đoán (Precision)

- Công thức:

$$\text{Precision} = \frac{\text{Số lượng dự đoán đúng (True Positive)}}{\text{Tổng số lượng dự đoán là True (True Positive + False Positive)}}$$

- Precision: Giá trị được in ra từ kết quả kiểm thử là **X.XX**, phản ánh tỷ lệ các dự đoán chính xác so với tổng số dự đoán dương tính.

3.5.2. Độ nhạy (Recall)

- Công thức:

$$\text{Recall} = \frac{\text{Số lượng dự đoán đúng (True Positive)}}{\text{Tổng số lượng thực tế cần nhận diện (True Positive + False Negative)}}$$

- Recall là một chỉ số quan trọng để đánh giá mức độ hiệu quả của mô hình trong việc nhận diện đúng các khuôn mặt. Với kết quả Recall đạt được, mô hình cho thấy khả năng nhận diện tốt, phù hợp để ứng dụng vào thực tế.

3.5.3. F1-Score

- Công thức:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- F1-Score là một chỉ số tổng hợp giữa độ chính xác (Precision) và độ thu hồi (Recall), phản ánh khả năng của mô hình trong việc duy trì một sự cân bằng giữa việc nhận diện chính xác các khuôn mặt và khả năng tìm kiếm được tất cả các khuôn mặt trong dữ liệu.
- Kết quả F1-Score của mô hình nhận diện khuôn mặt là **xx.xx**, cho thấy mô hình có khả năng duy trì sự cân bằng giữa việc nhận diện chính xác khuôn mặt và khả năng nhận diện đúng các khuôn mặt trong bộ dữ liệu kiểm thử. F1-Score cao chứng tỏ mô hình không chỉ chính xác mà còn có khả năng nhận diện đầy đủ các khuôn mặt mà không bỏ sót.

3.5.4. Độ chính xác tổng thể (Accuracy)

- Công thức:

$$\text{Accuracy} = \frac{\text{Số dự đoán đúng}}{\text{Tổng số dự đoán}}$$

- Kết quả đánh giá cho thấy độ chính xác (accuracy) của mô hình là **xx.xx%**, có nghĩa là mô hình đã dự đoán đúng tên của các khuôn mặt trong **xx.xx%** tổng số trường hợp trong bộ kiểm thử.

3.5.5. Confusion Matrix

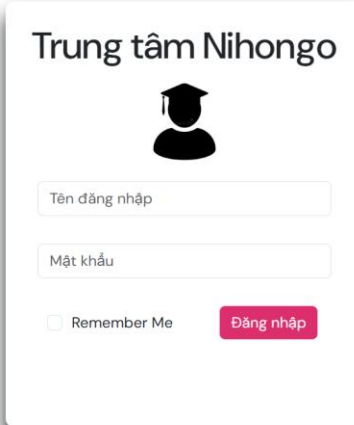
- Confusion Matrix là một công cụ quan trọng để đánh giá hiệu suất của mô hình phân loại, đặc biệt trong các bài toán phân loại có nhiều lớp (multi-class classification). Confusion Matrix cung cấp cái nhìn chi tiết về cách mà các dự đoán của mô hình phân loại so với nhãn thật.
- Confusion Matrix là một bảng 2x2 (cho bài toán phân loại nhị phân) hoặc một bảng NxN (cho bài toán phân loại đa lớp). Với bài toán phân loại nhị phân, bảng confusion matrix bao gồm 4 thành phần:

Bảng 3. Bảng thành phần Confusion Matrix

	Dự đoán là Positive	Dự đoán là Negative
Thực tế là Positive	True Positive (TP)	False Negative (FN)
Thực tế là Negative	False Positive (FP)	True Negative (TN)

3.6. Giao diện web

- Giao diện đăng nhập được thể hiện qua *hình 10*:



The image shows a login form for 'Trung tâm Nihongo'. At the top, the text 'Trung tâm Nihongo' is displayed above a silhouette of a person wearing a graduation cap. Below this, there are two input fields: 'Tên đăng nhập' (Username) and 'Mật khẩu' (Password). Under the password field, there is a checkbox labeled 'Remember Me' and a red button labeled 'Đăng nhập' (Login).

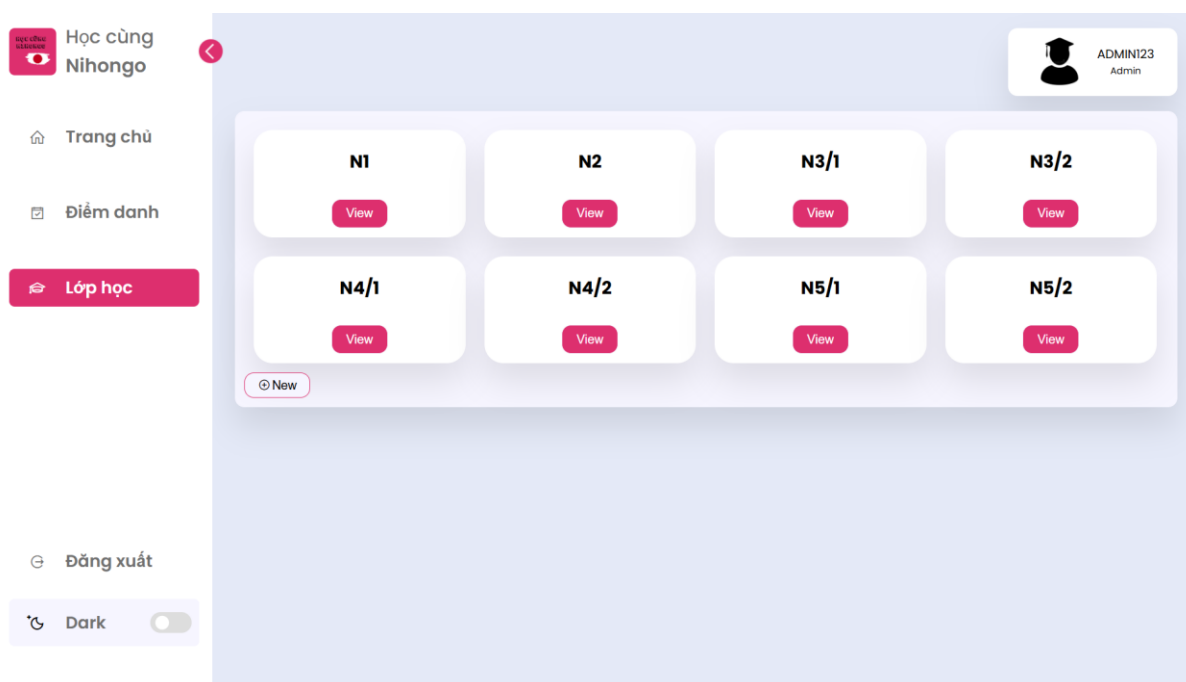
Hình 10. Giao diện đăng nhập

- Giao diện trang chủ được thể hiện qua hình 11:



Hình 11. Giao diện trang chủ

- Giao diện tổng quan lớp học được thể hiện qua hình 12:



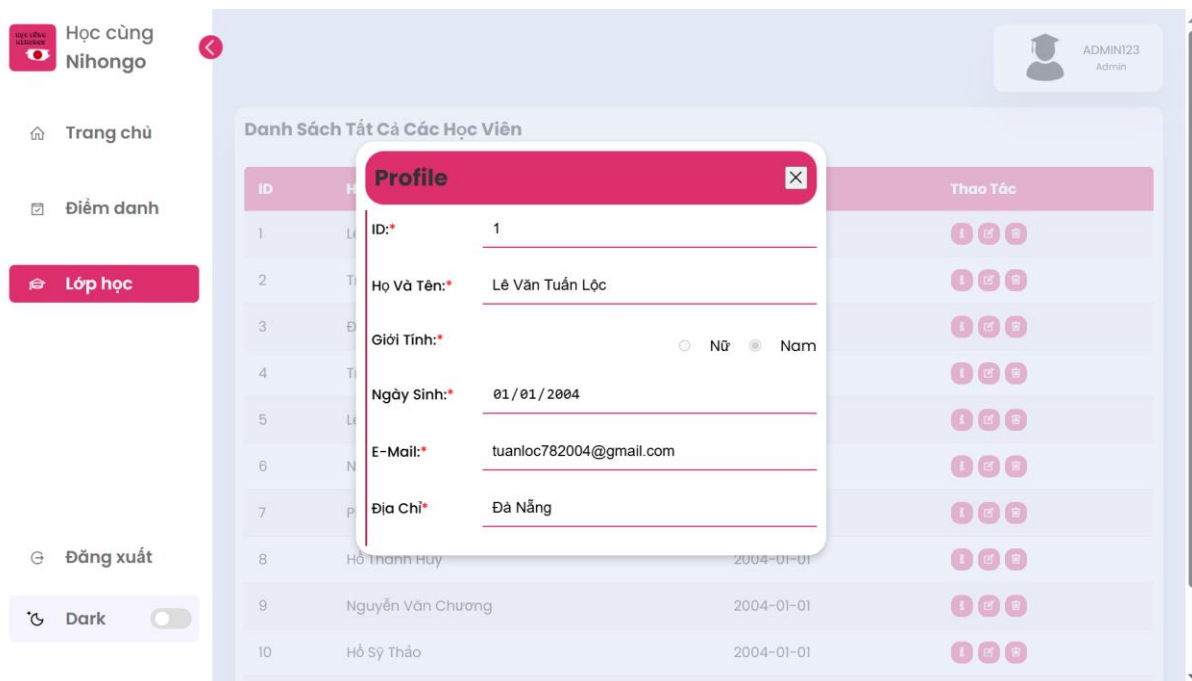
Hình 12. Giao diện tổng quan lớp học

- Giao diện học viên trong một lớp học được thể hiện qua hình 13:



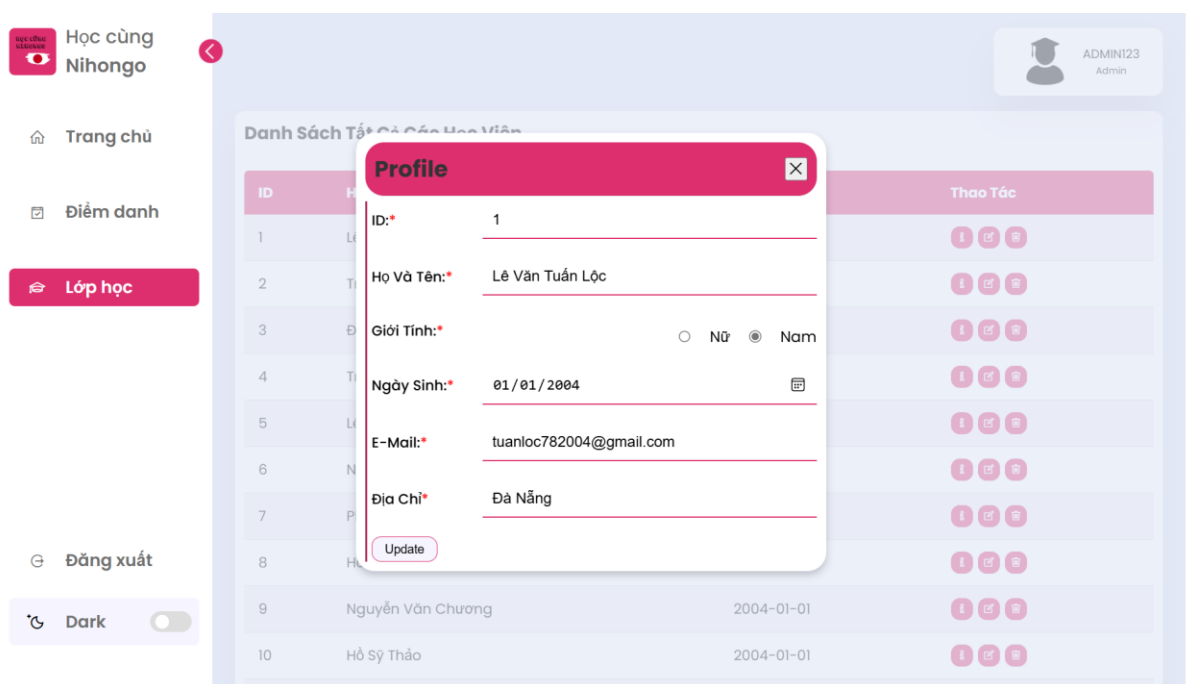
Hình 13. Giao diện học viên trong một lớp học

- Giao diện xem thông tin chi tiết học viên được thể hiện qua hình 14:



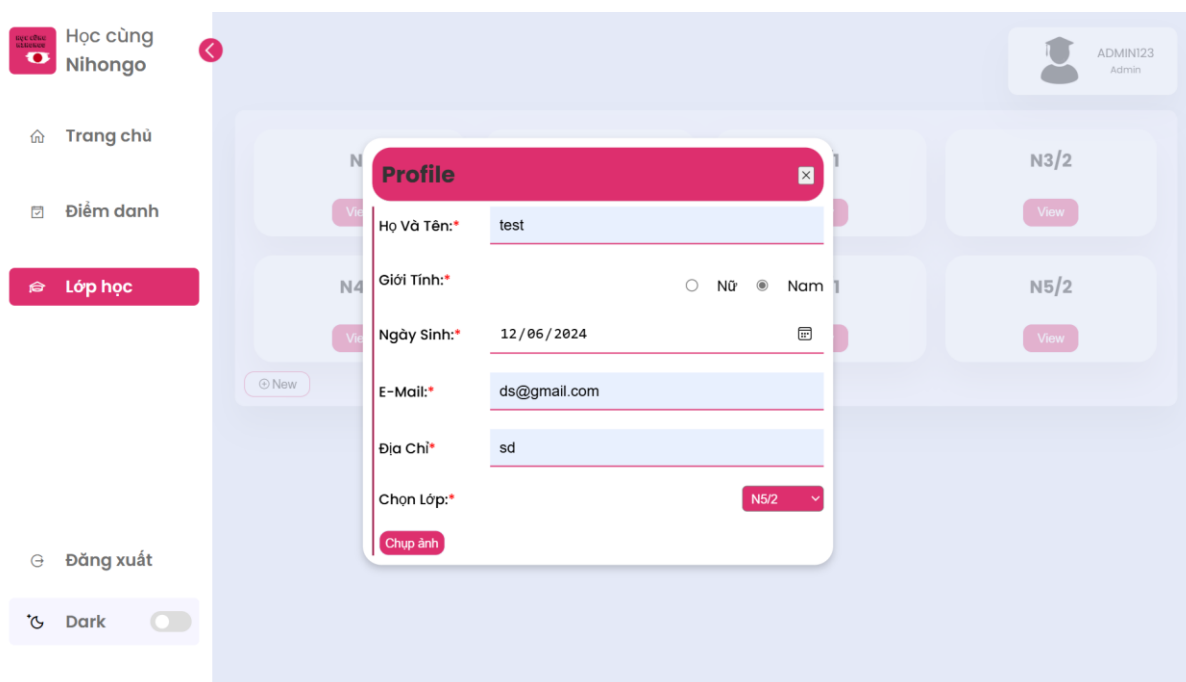
Hình 14. Giao diện xem thông tin chi tiết học viên

- Giao diện sửa thông tin học viên được thể hiện qua hình 15:



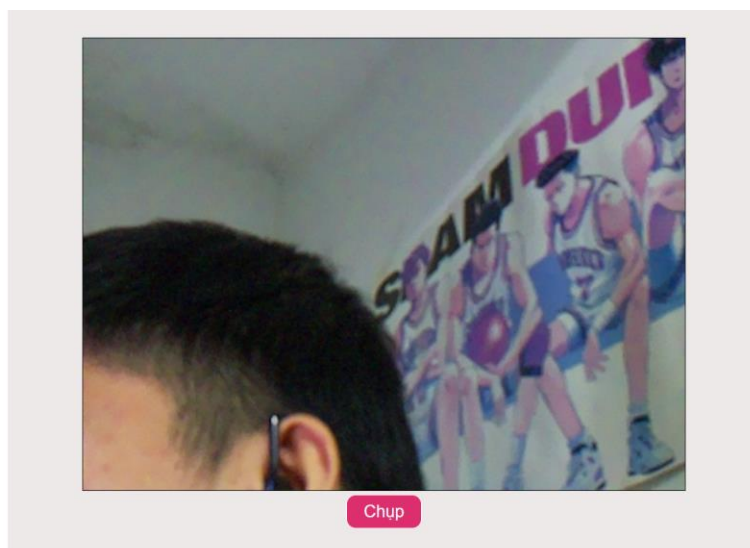
Hình 15. Giao diện sửa thông tin học viên

- Giao diện thêm học viên mới được thể hiện qua hình 16:



Hình 16. Giao diện thêm học viên mới

- Giao diện chụp ảnh khi thêm học viên mới được thể hiện qua *hình 17*:



Hình 17. Giao diện chụp ảnh khi thêm học viên

- Giao diện lịch học trong ngày được thể hiện qua *hình 18*:

Học cùng Nihongo

Trang chủ

Điểm danh

Lớp học

Đăng xuất

Dark

ADMINI23 Admin

Lịch Học

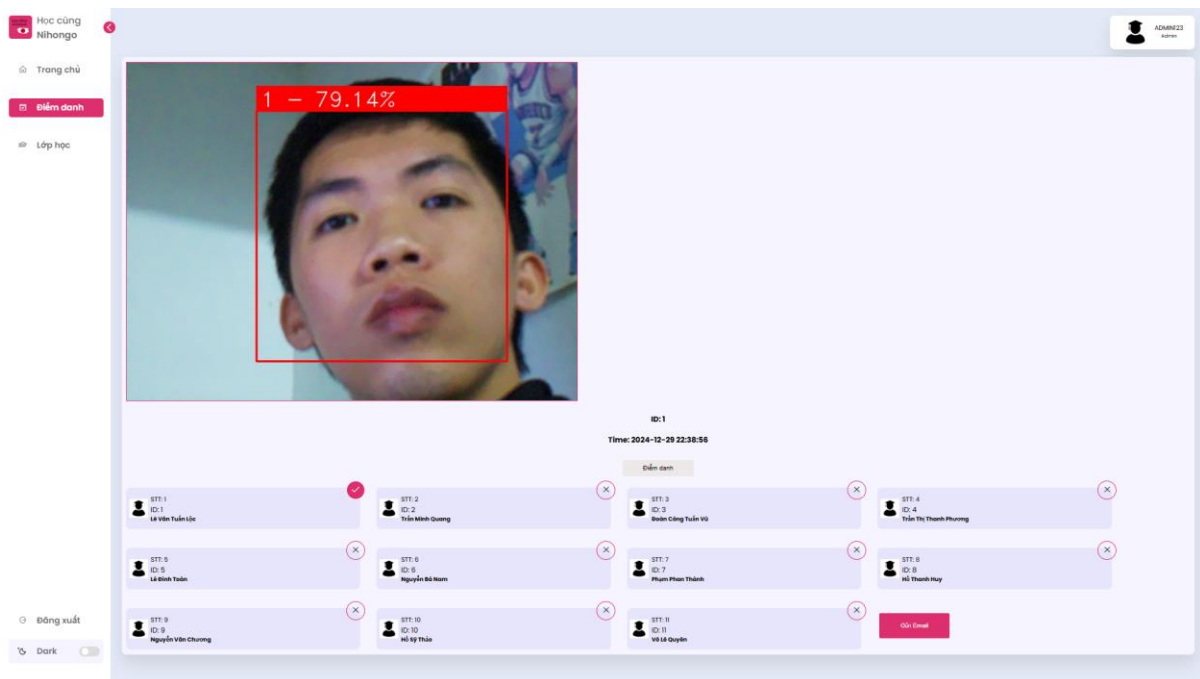
Ngày Học 12/29/2024

Điểm danh

STT	Lớp	Giờ Bắt Đầu	Giờ Kết Thúc	Điểm Danh
1	N1	07:00:00	23:00:00	
2	N4/1	09:00:00	23:00:00	
3	N5/2	15:30:00	17:20:00	

Hình 18. Giao diện lịch học trong ngày

- Giao diện điểm danh được thể hiện qua hình 19:



Hình 19. Giao diện điểm danh

- Email thông báo cho học sinh đã điểm danh được thể hiện qua hình 20:

Admin <phongsuga333@gmail.com>

đến tôi ▾

 THÔNG BÁO

Kính gửi học viên,

Hôm nay, bạn đã có mặt trong buổi học.

Chúc bạn học tập hiệu quả và đạt nhiều thành tích tốt!

Trân trọng,

[Ban giám hiệu/ Giáo viên phụ trách]

Hình 20. Email điểm danh

- Email thông báo cho học sinh đã vắng mặt được thể hiện qua *hình 21*:

Admin <phungsuga333@gmail.com>

22:39

đến toanledinh76 ▼

 THÔNG BÁO

Kính gửi học viên,

Hôm nay, bạn đã vắng mặt trong buổi học.

Vui lòng đến lớp vào ngày học tiếp theo.

Nếu có lý do chính đáng, vui lòng liên hệ giáo viên phụ trách để được hỗ trợ.

Trân trọng,

[Ban giám hiệu/ Giáo viên phụ trách]

Hình 21. Email vắng mặt

4. Kết luận

Dự án hệ thống nhận diện khuôn mặt thông minh đã đạt được mục tiêu đề ra, bao gồm tự động hóa quy trình điểm danh và hỗ trợ quản lý học sinh hiệu quả tại các trung tâm ngoại ngữ. Với việc ứng dụng công nghệ AI, IoT cùng các thư viện OpenCV và Face Recognition, hệ thống đảm bảo khả năng nhận diện khuôn mặt nhanh chóng, chính xác và thân thiện với người dùng. Dữ liệu nhận diện được lưu trữ và xử lý trên server, hỗ trợ gửi thông báo qua email và cung cấp các báo cáo trực quan.

Trong tương lai, hệ thống có thể được nâng cấp bằng cách tối ưu hóa thuật toán để tăng độ chính xác, cải thiện bảo mật nhằm bảo vệ dữ liệu người dùng, phát triển thêm các tính năng nâng cao như báo cáo phân tích chi tiết hoặc hỗ trợ đa nền tảng. Với các hướng phát triển này, hệ thống sẽ mở rộng khả năng ứng dụng không chỉ trong giáo dục mà còn trong các lĩnh vực khác như quản lý doanh nghiệp hay an ninh.

Kết quả đạt được từ đề án là tiền đề quan trọng để tiếp tục nghiên cứu và phát triển các giải pháp công nghệ số, góp phần hiện đại hóa quy trình quản lý và nâng cao trải nghiệm của người dùng trong thời đại chuyển đổi số.

5. Danh mục tài liệu tham khảo

- [1] <https://core-electronics.com.au/guides/raspberry-pi/>
- [2] B. Heisele, P. Ho, and T. Poggio, “Face recognition with support vector machines: Global versus component-based approach,” in International Conference on Computer Vision (ICCV'01), 2001.
- [3] <https://pypi.org/project/face-recognition/>
- [4] <https://github.com/opencv/opencv>
- [5] https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API
- [6] K. C. Yow, R. Cipolla, “Feature - Based Human Face Detection”, Image and Vision Computing, Vol. 15, No. 9, pp. 713-735, 1997.
- [7] I. Craw, D. Tock, A. Bennett, “Finding Face Features”, Proc. 2nd European Conf. Computer Vision (ECCV'92), Vol. 2, pp. 92-96, 1992.
- [8] A. Lanitis, C. J. Taylor, T. F. Cootes, “An Automatic Face Identification System Using Flexible Appearance Models”, Image and Vision Computing, Vol. 13, No. 5, pp. 393-401, 1995.
- [9] Phạm Thế Bảo, Nguyễn Thành Nhứt, Cao Minh Thịnh, Trần Anh Tuấn, Phan Phúc Doãn, “Tổng quan các phương pháp xác định khuôn mặt người”, Tạp chí Công nghệ thông tin & Truyền thông.