BÁO CÁO ĐỀ TÀI

# XÂY DỰNG HỆ THỐNG ĐIỂM DANH SINH VIÊN KẾT HỢP IoT VÀ AI

Nhóm thực hiện: .............................................

Giảng viên hướng dẫn: .....................................

Ngày nộp: ................................................

# 1. Giới thiệu đề tài

Trong bối cảnh chuyển đổi số mạnh mẽ, việc tự động hóa quản lý điểm danh sinh viên là nhu cầu cấp thiết nhằm nâng cao hiệu quả, giảm gian lận và tiết kiệm nguồn lực. Đề tài này xây dựng hệ thống điểm danh sinh viên ứng dụng AI (nhận diện khuôn mặt) kết hợp IoT, giúp quản lý tập trung, minh bạch và thuận tiện cho cả nhà trường lẫn sinh viên. Hệ thống có thể mở rộng cho nhiều lớp học, nhiều điểm điểm danh, tích hợp với các thiết bị IoT như cửa tự động, đèn báo, ...

Việc điểm danh truyền thống vốn dựa vào sổ sách hoặc ký tên trực tiếp không chỉ tốn thời gian mà còn dễ xảy ra sai sót và gian lận. Với quy mô lớp học ngày càng lớn và yêu cầu quản lý minh bạch, một hệ thống tự động, chính xác sẽ giúp tiết kiệm thời gian, công sức cho giảng viên và nhà trường, đồng thời nâng cao ý thức chấp hành của sinh viên.

Hệ thống sử dụng các công nghệ hiện đại như camera AI, máy tính nhỏ gọn (Raspberry Pi) và cơ sở dữ liệu NoSQL (MongoDB), đảm bảo lưu trữ và xử lý thông tin nhanh chóng, an toàn. Nhờ tận dụng nền tảng mã nguồn mở và thiết bị sẵn có, hệ thống có tính linh hoạt cao, dễ dàng triển khai ở nhiều phòng học hoặc cơ sở giáo dục khác nhau mà không yêu cầu đầu tư lớn.

Ngoài ra, việc tích hợp các thiết bị IoT như cảm biến chuyển động, cửa tự động hoặc hệ thống đèn báo không chỉ giúp điểm danh trở nên tự động hóa hoàn toàn mà còn nâng cao mức độ an toàn và kiểm soát. Khi sinh viên điểm danh thành công, hệ thống có thể gửi tín hiệu điều khiển các thiết bị thông minh, mở rộng khả năng tương tác giữa hệ thống quản lý và môi trường thực tế.

Với giao diện web trực quan được xây dựng trên nền tảng Flask và Bootstrap, quản trị viên có thể dễ dàng thêm mới sinh viên, cập nhật thông tin, theo dõi lịch sử điểm danh và xuất báo cáo thống kê chi tiết. Sinh viên cũng có thể được cung cấp quyền truy cập để kiểm tra lịch sử điểm danh của mình, đảm bảo tính minh bạch và công bằng.

Đề tài không chỉ mang tính ứng dụng thực tiễn mà còn mở ra tiềm năng nghiên cứu và phát triển thêm nhiều tính năng mới trong tương lai. Với nền tảng đã xây dựng, nhóm thực hiện có thể tích hợp thêm các công nghệ khác như nhận diện giọng nói, quét QR code, hoặc hệ thống cảnh báo sớm khi phát hiện người lạ, nhằm hướng tới một hệ thống quản lý thông minh, toàn diện và linh hoạt.

**2. Thành phần linh kiện sử dụng**

## 2.1. Phần cứng (IoT)

- Camera/Webcam: Thu hình khuôn mặt sinh viên tại điểm danh, có thể dùng camera USB hoặc camera tích hợp trên máy tính/laptop.

- Máy tính/Raspberry Pi: Thiết bị xử lý trung tâm, nhận dữ liệu từ camera, chạy phần mềm nhận diện và quản lý dữ liệu.

- (Tùy chọn) Thiết bị IoT khác: Ví dụ: cửa tự động, đèn báo, cảm biến chuyển động, RFID để tăng tính bảo mật và tự động hóa.

## 2.2. Phần mềm (AI & Backend)

- Python: Ngôn ngữ lập trình chính, dễ phát triển và tích hợp các thư viện AI, IoT.

- Flask: Framework web nhẹ, dễ triển khai giao diện quản trị và API.

- OpenCV: Thư viện xử lý ảnh mạnh mẽ, hỗ trợ thao tác với camera, tiền xử lý ảnh.

- face\_recognition: Thư viện AI nhận diện khuôn mặt dựa trên deep learning, cho độ chính xác cao.

- MongoDB: Cơ sở dữ liệu NoSQL, lưu trữ linh hoạt thông tin sinh viên, ảnh mẫu, lịch sử điểm danh.

- Bootstrap: Thư viện giao diện web hiện đại, responsive, thân thiện với người dùng.

# 3. Kinh phí dự kiến

|  |  |  |  |
| --- | --- | --- | --- |
| Thành phần | Số lượng | Đơn giá (VNĐ) | Thành tiền (VNĐ) |
| Camera HD | 1 | 400.000 | 400.000 |
| Máy tính/Raspberry Pi | 1 | 2.000.000 | 2.000.000 |
| (Tùy chọn) RFID | 0 | 0 | 0 |
| Chi phí điện, mạng | - | 200.000 | 200.000 |
| Tổng cộng |  |  | 2.600.000 |

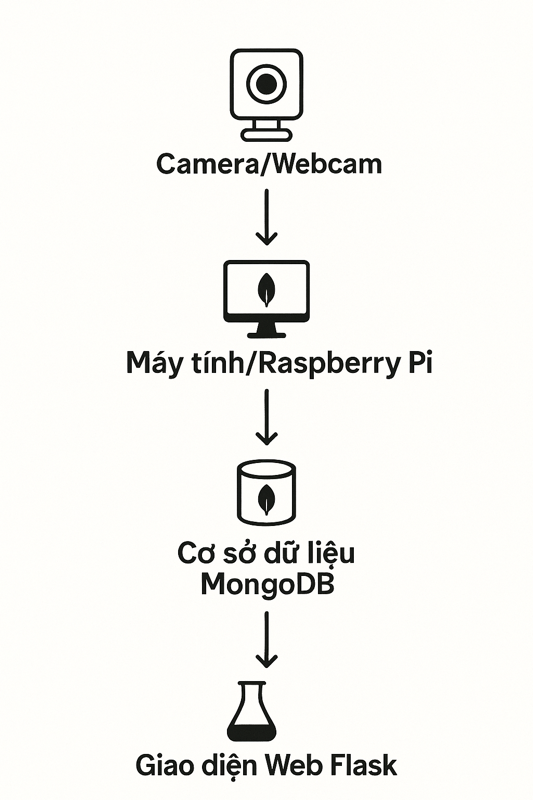
\*Lưu ý: Nếu dùng máy tính cá nhân, chi phí phần cứng có thể giảm. Hệ thống có thể mở rộng với nhiều camera hoặc thiết bị IoT khác nếu cần.

# 4. Thời gian lắp đặt

|  |  |
| --- | --- |
| Công việc | Thời gian (ngày) |
| Nghiên cứu, chuẩn bị linh kiện | 2 |
| Lắp đặt phần cứng, kết nối camera | 1 |
| Cài đặt phần mềm, môi trường | 1 |
| Lập trình, tích hợp hệ thống | 2 |
| Kiểm thử, hoàn thiện | 1 |
| Tổng cộng | 7 |

# 5. Sơ đồ khối & liên kết hệ thống

## 5.1. Sơ đồ khối tổng thể



## 5.2. Mô tả liên kết các thành phần

- Camera/Webcam: Thu hình ảnh khuôn mặt sinh viên, truyền về máy tính/Raspberry Pi.

- Máy tính/Raspberry Pi: Nhận hình ảnh từ camera, xử lý nhận diện khuôn mặt bằng AI, giao tiếp với MongoDB, chạy server Flask.

- MongoDB: Lưu trữ thông tin sinh viên, ảnh mẫu, lịch sử điểm danh.

- Giao diện Web: Quản lý sinh viên, xem lịch sử điểm danh, xem trực tiếp camera.

- (Tùy chọn) Các thiết bị IoT khác: Có thể tích hợp để tự động mở cửa, bật đèn, gửi thông báo khi sinh viên điểm danh thành công.

# 6. Mô tả hoạt động hệ thống

1. Thêm sinh viên: Quản trị viên thêm sinh viên mới qua web, upload/chụp ảnh khuôn mặt. Ảnh này sẽ được dùng làm mẫu nhận diện.

2. Điểm danh: Camera ghi lại hình ảnh sinh viên, AI nhận diện khuôn mặt, ghi nhận điểm danh vào MongoDB. Có thể tích hợp cảnh báo nếu phát hiện người lạ.

3. Quản lý: Quản trị viên xem, chỉnh sửa thông tin sinh viên, xuất lịch sử điểm danh, thống kê tỷ lệ chuyên cần.

4. Mở rộng: Có thể tích hợp thêm các chức năng như gửi email thông báo, kết nối hệ thống quản lý đào tạo, ...

# 7. Giải thích thuật toán nhận diện khuôn mặt

**Quy trình nhận diện khuôn mặt trong hệ thống gồm các bước chính sau:**

1. Tiền xử lý ảnh:

Ảnh thu được từ camera thường là ảnh màu (RGB). Để đơn giản hóa xử lý, ảnh sẽ được chuyển sang ảnh đen trắng (grayscale). Việc này giúp giảm nhiễu, giảm số chiều dữ liệu và tăng tốc độ xử lý.

2. Phát hiện vùng khuôn mặt:

Sử dụng các mô hình phát hiện khuôn mặt (ví dụ: HOG, CNN hoặc Haar Cascade) để xác định vị trí các khuôn mặt trong ảnh. Kết quả là các bounding box bao quanh từng khuôn mặt.

3. Căn chỉnh và cắt khuôn mặt:

Ảnh khuôn mặt được cắt ra từ ảnh gốc, có thể căn chỉnh lại để chuẩn hóa kích thước, hướng nhìn, ... giúp tăng độ chính xác khi nhận diện.

4. Trích xuất đặc trưng khuôn mặt:

Sử dụng các mô hình deep learning (ví dụ: FaceNet, dlib, ResNet) để chuyển ảnh khuôn mặt thành vector đặc trưng (embedding). Mỗi khuôn mặt sẽ có một vector đặc trưng duy nhất, phản ánh các đặc điểm sinh trắc học của từng người.

**5. So sánh đặc trưng và nhận diện:**

- Vector đặc trưng của khuôn mặt mới sẽ được so sánh với các vector đã lưu trong cơ sở dữ liệu (dùng khoảng cách Euclidean hoặc cosine). Nếu khoảng cách nhỏ hơn ngưỡng cho phép, hệ thống xác định đó là cùng một người và ghi nhận điểm danh.

**6. Lưu trữ và cập nhật dữ liệu:**

- Nếu nhận diện thành công, hệ thống sẽ lưu lại thông tin điểm danh (thời gian, mã sinh viên, ...) vào cơ sở dữ liệu. Nếu không nhận diện được, có thể cảnh báo hoặc ghi nhận là người lạ.

## 7.1. Đầu vào và đầu ra của thuật toán

- Đầu vào: Ảnh/video từ camera chứa khuôn mặt sinh viên (ảnh màu RGB).

- Đầu ra: Kết quả nhận diện (ID sinh viên nếu nhận diện thành công hoặc thông báo "Người lạ" nếu không có kết quả).

## 7.2. Hoạt động tốt khi nào?

- Ánh sáng đầy đủ, khuôn mặt rõ nét, không che khuất.

- Dữ liệu mẫu lưu trong database chất lượng cao (nhiều góc chụp, ít nhiễu).

- Camera chất lượng ổn định, độ phân giải đủ cao (HD trở lên).

## 7.3. Hoạt động không tốt khi nào?

- Ảnh mờ, nhiễu, thiếu sáng.

- Khuôn mặt bị che (khẩu trang, kính râm, tóc che…).

- Cơ sở dữ liệu mẫu không đầy đủ hoặc bị lỗi.

## 7.4. Nguyên nhân gây sai lệch

- Model không nhận diện tốt nếu ánh sáng kém hoặc góc chụp không chính diện.

- Số lượng ảnh mẫu trong database quá ít, không đủ đa dạng góc độ.

- Camera chất lượng thấp làm giảm độ phân giải, khiến feature extraction kém chính xác.

**8. Kết luận:**

Qua quá trình nghiên cứu và xây dựng hệ thống, chúng ta đã thiết kế thành công một mô hình giám sát dữ liệu sử dụng Camera/Webcam, xử lý bởi Máy tính hoặc Raspberry Pi, lưu trữ dữ liệu trong cơ sở dữ liệu MongoDB, và hiển thị dữ liệu trên Giao diện Web Flask. Hệ thống này đáp ứng được các yêu cầu cơ bản về thu thập, xử lý và hiển thị dữ liệu một cách trực quan, đồng thời đảm bảo tính ổn địƯnh và dễ mở rộng.

Kết quả đạt được chứng minh tính khả thi của việc tích hợp các công nghệ mã nguồn mở như Flask, MongoDB, cùng với các thiết bị IoT phổ biến để xây dựng các giải pháp giám sát thông minh. Hệ thống không chỉ hỗ trợ người dùng trong việc quản lý dữ liệu mà còn tạo nền tảng để phát triển thêm các tính năng nâng cao như phân tích dữ liệu thời gian thực, cảnh báo tự động và tích hợp các mô hình học máy.

Tuy nhiên, để tối ưu hóa hiệu suất, trong các nghiên cứu tiếp theo, có thể xem xét việc nâng cấp phần cứng (như sử dụng Raspberry Pi mạnh hơn hoặc máy tính công suất cao hơn), tối ưu hóa truy vấn cơ sở dữ liệu, và tăng cường tính bảo mật của hệ thống. Qua đó, hệ thống sẽ trở thành một công cụ hữu ích, ứng dụng tốt trong nhiều lĩnh vực như an ninh, công nghiệp, và nghiên cứu khoa học.