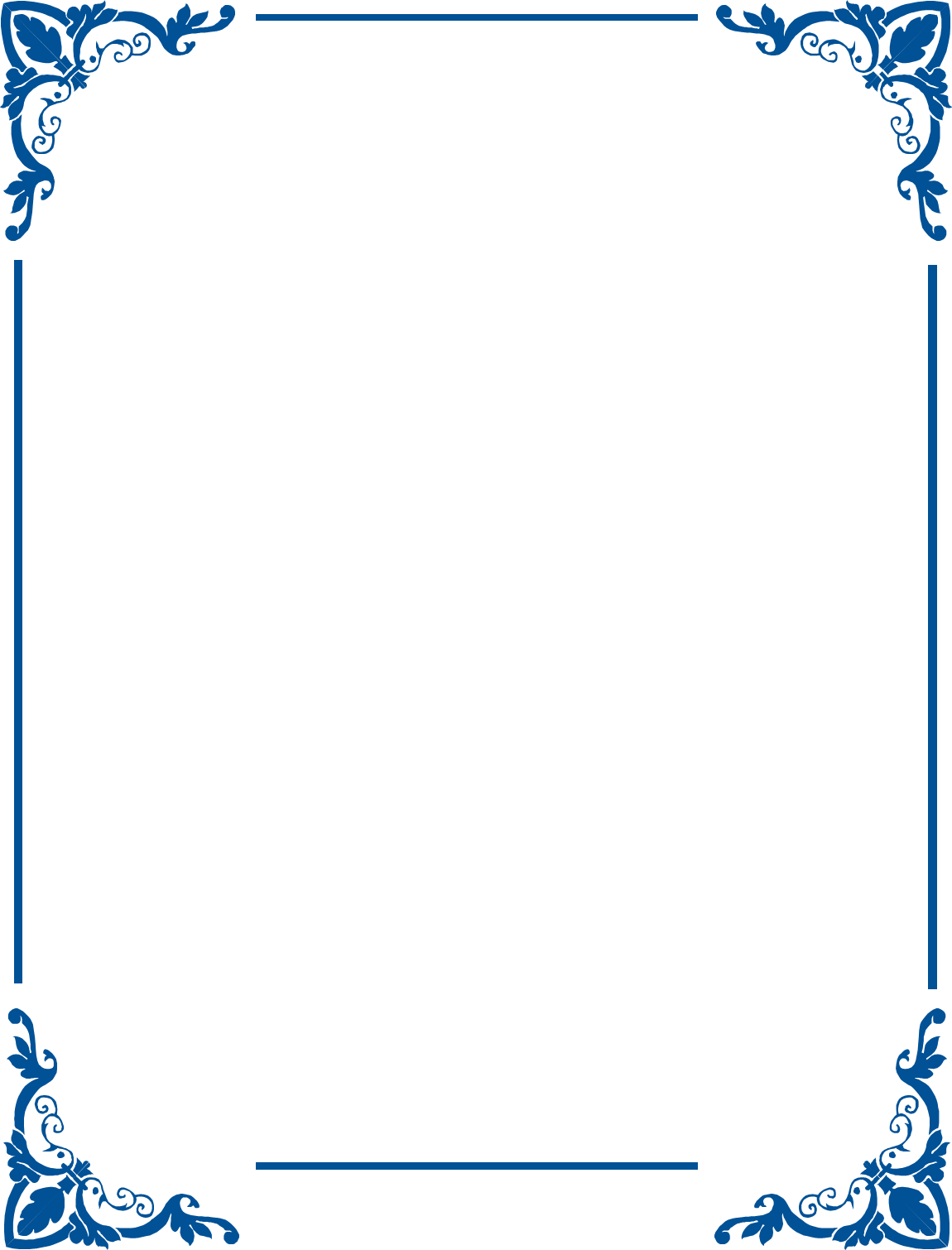
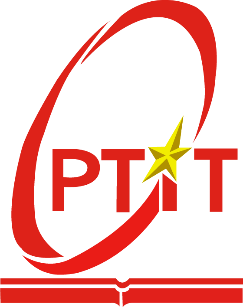
****

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

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

**CƠ SỞ TẠI THÀNH PHỐ HỒ CHÍ MINH**

**------------oO🕮Oo------------**



**BÁO CÁO CUỐI KÌ**

**MÔN HỌC: IOT VÀ ỨNG DỤNG**

Đề tài: **“NHẬN DIỆN KHUÔN MẶT DỰA VÀO HỌC MÁY”**

|  |  |  |
| --- | --- | --- |
| **Giảng viên** | **:** | **Đàm Minh Lịch** |
| **Sinh viên thực hiện** | **:** | **Lê Văn Phúc - N20DCCN128**  **Nguyễn Đắc Xuân Trung – N20DCCN159** |
| **Lớp** | **:** | **D20CQCNPM02-N** |

**TP. Hồ Chí Minh, ngày 4 tháng 1 năm 2024**

# LỜI CẢM ƠN

Lời đầu tiên, em xin chân thành cảm ơn đến thầy ***Đàm Minh Lịnh***. Trong quá trình học môn *IoT và ứng dụng*, nhờ nhận được sự hướng dẫn và giúp đỡ tận tình của thầy, em đã biết cách sử dụng các thiết bị IoT và nhúng mã vào chúng. Đồng thời, em cũng đã tích lũy được những kiến thức và kỹ năng cần thiết để hỗ trợ cho việc xây dựng và phát triển các phần mềm liên quan khác.

Bên cạnh đó, chúng em cũng xin cảm ơn Ban lãnh đạo và các giảng viên tại Học Viện Công Nghệ Bưu Chính Viễn Thông đã tạo điều kiện, cơ sở vật chất để chúng em có cơ hội và môi trường học tập và rèn luyện.

Tuy nhiên, với sự thiếu sót về kiến thức và kinh nghiệm cũng như thời gian có hạn nên khó tránh những hạn chế, sai sót, em mong nhận được sự đóng góp ý kiến của thầy để em có thể học hỏi thêm được những kiến thức và kinh nghiệm để có thể hoàn thiện đề tài một cách tốt nhất.

Em xin trân trọng cảm ơn!

Thành phố Hồ Chí Minh, tháng 1 năm 2024

**Sinh viên thực hiện**

Lê Văn Phúc

Nguyễn Đắc Xuân Trung

# MỞ ĐẦU

Trong thời đại hiện đại, sự hội nhập giữa trí tuệ nhân tạo và học máy đã mở ra những khả năng đáng kinh ngạc, đặc biệt là trong lĩnh vực nhận diện khuôn mặt. Với sự phát triển nhanh chóng của công nghệ, việc áp dụng học máy để nhận diện khuôn mặt không chỉ mang lại những ứng dụng hấp dẫn mà còn mở ra cánh cửa cho những khả năng đặc sắc trong tương tác máy tính - con người.

Đề tài này tập trung vào nghiên cứu và ứng dụng của mô hình nhận diện khuôn mặt dựa trên học máy. Điều này không chỉ đòi hỏi sự kết hợp sáng tạo giữa trí tuệ nhân tạo và công nghệ học máy, mà còn đặt ra những thách thức quan trọng trong việc hiểu rõ và đáp ứng chính xác đối với độ phức tạp của khuôn mặt con người.

Với việc tập trung vào việc nhận diện khuôn mặt, chúng ta sẽ khám phá sự mạnh mẽ của học máy trong việc xử lý và phân tích hình ảnh khuôn mặt, từ đó đưa ra những kết quả đáng kể. Bằng cách sử dụng mô hình được đào tạo trên dữ liệu chất lượng, chúng tôi hy vọng mở rộng ứng dụng của công nghệ nhận diện khuôn mặt từ bảo mật và an ninh đến các lĩnh vực như tương tác người dùng, y tế, và giáo dục.

Qua nghiên cứu này, chúng tôi mong muốn góp phần vào sự hiểu biết và tiến bộ trong lĩnh vực nhận diện khuôn mặt dựa vào học máy, mở ra những triển vọng mới cho việc tận dụng tiềm năng của công nghệ để cải thiện chất lượng cuộc sống hàng ngày và giải quyết những thách thức đa dạng của xã hội hiện đại.

# MỤC LỤC

[LỜI CẢM ƠN 2](#_Toc155122198)

[MỞ ĐẦU 3](#_Toc155122199)

[MỤC LỤC 4](#_Toc155122200)

[DANH MỤC HÌNH ẢNH 5](#_Toc155122201)

[PHẦN I: GIỚI THIỆU 6](#_Toc155122202)

[1. Giới thiệu đề tài 6](#_Toc155122203)

[2. Mục tiêu đề tài 6](#_Toc155122204)

[3. Công cụ lập trình 6](#_Toc155122205)

[4. Giới thiệu ngôn ngữ lập trình Python 7](#_Toc155122206)

[5. Các thư viện – Framework hổ trợ 8](#_Toc155122207)

[PHẦN II: CƠ SỞ LÝ THUYẾT 9](#_Toc155122208)

[1. Phát hiện khuôn mặt bằng mô hình Haar Cascade Classifier: 9](#_Toc155122209)

[2. Nhận dạng khuôn mặt - Face Recognition 11](#_Toc155122210)

[PHẦN III: XÂY DỰNG CHƯƠNG TRÌNH 12](#_Toc155122211)

[1. Xây dựng mô hình Face Recognition 12](#_Toc155122212)

[2. Các bước thực hiện 13](#_Toc155122213)

[3. Cấu hình ESP32-CAM 15](#_Toc155122214)

[PHẦN IV: KẾT QUẢ 17](#_Toc155122215)

[PHẦN V: TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN 20](#_Toc155122216)

[1. Tổng kết: 20](#_Toc155122217)

[2. Hướng phát triển: 21](#_Toc155122218)

[PHỤ LỤC 21](#_Toc155122219)

[Works Cited 24](#_Toc155122220)

[TÀI LIỆU THAM KHẢO 24](#_Toc155122221)

# DANH MỤC HÌNH ẢNH

[Hình 1.1: Thiết bị ESP32-CAM 7](#_Toc155122222)

[Hình 2.1: Một số đặc điểm haar 9](#_Toc155122223)

[Hình 2.2: Đặc trưng haar trên khuôn mặt 11](#_Toc155122224)

[Hình 3.1: Sơ đồ các bước nhận dạng bằng face-recognition 12](#_Toc155122225)

[Hình 3.1: Quy trình nhận diện khuôn mặt 13](#_Toc155122226)

[Hình 3.2: Xác định khuôn mặt trong ảnh 13](#_Toc155122227)

[Hình 3.3: Trích xuất đặc trưng ảnh 14](#_Toc155122228)

[Hình 3.4: Các ảnh có đặc trưng gần giống nhau 14](#_Toc155122229)

[Hình 3.5: Phân loại dựa vào khoảng cách embeddings 15](#_Toc155122230)

[Hình 4.1: Giao diện 17](#_Toc155122231)

[Hình 4.2: Lấy ảnh mẫu 18](#_Toc155122232)

[Hình 4.3: Camera nhận diện các khuôn mặt 19](#_Toc155122233)

[Hình 4.4: Thông báo sỉ số 19](#_Toc155122234)

[Hình 4.5: Lưu thông tin sinh viên có mặt 20](#_Toc155122235)

# PHẦN I: GIỚI THIỆU

## Giới thiệu đề tài

Đề tài "Nhận diện khuôn mặt dựa vào học máy" là một nghiên cứu tiên tiến trong lĩnh vực ứng dụng công nghệ nhận diện hình ảnh để tự động hóa quá trình điểm danh và kiểm soát sự hiện diện cá nhân. Bằng cách sử dụng các thuật toán và mô hình máy học tiên tiến, đề tài này tập trung vào việc phát triển hệ thống có khả năng nhận diện độ chính xác cao, linh hoạt và an toàn.

Nhận diện khuôn mặt không chỉ mang lại sự thuận tiện trong quá trình điểm danh mà còn giúp giảm thời gian và công sức so với các phương pháp truyền thống. Điều này làm cho đề tài trở thành một nguồn động lực quan trọng trong việc nghiên cứu và áp dụng các công nghệ mới nhằm tối ưu hóa quản lý sự hiện diện trong các tổ chức, doanh nghiệp, trường học và các ngữ cảnh khác trong cuộc sống hàng ngày của chúng ta [1].

Tính nhạy cảm về quyền riêng tư và an toàn dữ liệu được xem xét kỹ lưỡng trong quá trình phát triển, đảm bảo rằng hệ thống không chỉ mang lại hiệu suất cao mà còn duy trì tính bảo mật và tuân thủ nghiêm ngặt các quy chuẩn an toàn thông tin. Đề tài này đại diện cho sự hội nhập giữa sự tiên tiến trong lĩnh vực trí tuệ nhân tạo và ứng dụng thực tế, với hy vọng tạo ra những đóng góp quan trọng cho việc cải thiện quá trình quản lý sự hiện diện và kiểm soát.

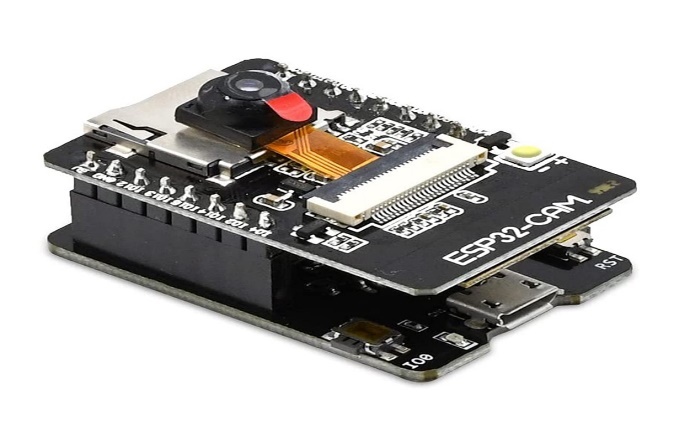
## Mục tiêu đề tài

Đề tài “Nhận diện khuôn mặt dựa vào học máy” được sử dụng trong các hệ thống:

* Điểm danh lớp học
* Chấm công nhân viên
* Phân tích dữ liệu

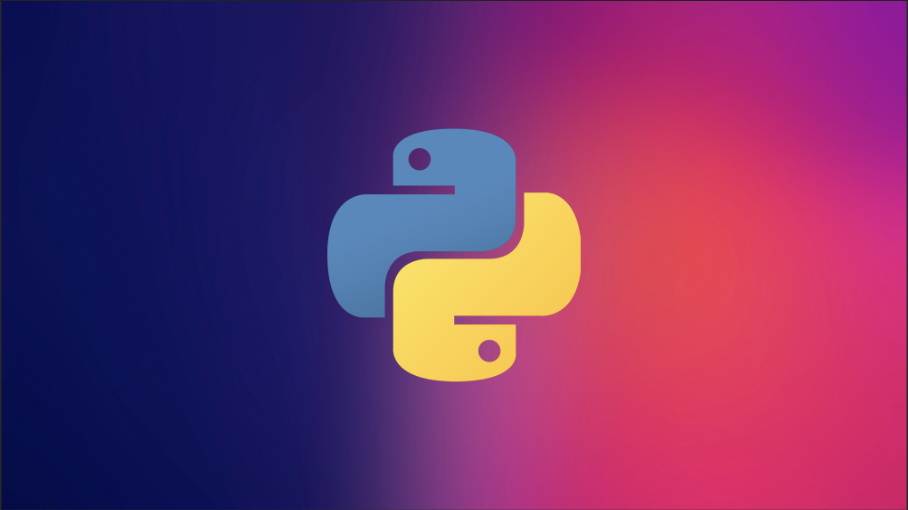
## Công cụ lập trình

* Công cụ lập trình: Pycharm (2023.2.3)
* Ngôn ngữ lập trình: Python
* Xây dụng giao diện: GUI sử dụng thư viện Tkinter
* "**ESP32-CAM**" là một module phổ biến được phát triển bởi Espressif Systems, sử dụng chip ESP32, một trong những vi xử lý đa nhiệm và có hiệu suất cao nhất trong dòng sản phẩm ESP8266/ESP32. Module này là sự kết hợp giữa vi xử lý mạnh mẽ và khả năng truyền thông không dây Wi-Fi, cùng với khả năng chụp ảnh và quay video, tạo ra một giải pháp linh hoạt cho các ứng dụng Internet of Things (IoT) và các dự án liên quan đến hình ảnh [2].



**Hình 1.1:** Thiết bị ESP32-CAM

## Giới thiệu ngôn ngữ lập trình Python



Python là một ngôn ngữ lập trình thông dịch, mạnh mẽ, và đa năng. Nó đã trở thành một trong những ngôn ngữ phổ biến nhất và được ưa chuộng trong ngành công nghiệp phần mềm, khoa học dữ liệu, trí tuệ nhân tạo, phát triển web, và nhiều lĩnh vực khác. Dưới đây là một số điểm quan trọng về Python:

Ngôn ngữ đọc dễ hiểu: Python thiết kế để có cú pháp đơn giản và dễ đọc, giúp người lập trình dễ dàng hiểu code của mình và của người khác. Điều này làm cho Python trở thành một ngôn ngữ tốt để bắt đầu lập trình.

Đa năng: Python có thư viện và khung làm việc với nhiều lĩnh vực khác nhau, bao gồm phát triển web (với Django và Flask), khoa học dữ liệu (với NumPy, Pandas, Matplotlib), trí tuệ nhân tạo (với TensorFlow, PyTorch), phát triển game (với Pygame), và nhiều ứng dụng khác.

Hệ sinh thái phong phú: Python có một cộng đồng mạnh mẽ và đa dạng, cung cấp hàng ngàn thư viện và framework khác nhau để hỗ trợ phát triển ứng dụng. Điều này giúp giảm thời gian phát triển và tận dụng được nhiều tài nguyên sẵn có.

Hỗ trợ đa nền tảng: Python chạy trên nhiều hệ điều hành, bao gồm Windows, macOS và các phiên bản khác của Unix/Linux. Điều này đồng nghĩa bạn có thể viết mã một lần và chạy trên nhiều nền tảng khác nhau mà không cần thay đổi mã nguồn.

Cộng đồng và tài liệu phong phú: Python có một cộng đồng lớn với hàng triệu người dùng và người lập trình. Có nhiều tài liệu, diễn đàn trực tuyến, và nguồn học tập để hỗ trợ người mới học và giải quyết vấn đề trong quá trình phát triển.

Phát triển web: Python hỗ trợ phát triển web với các framework như Django và Flask. Chúng cho phép bạn xây dựng các ứng dụng web mạnh mẽ và hiệu quả.

Khoa học dữ liệu và trí tuệ nhân tạo: Python đã trở thành một công cụ quan trọng trong lĩnh vực khoa học dữ liệu và trí tuệ nhân tạo, với thư viện mạnh mẽ như NumPy, Pandas, Matplotlib, TensorFlow và PyTorch.

Giới thiệu dành cho người mới học lập trình: Python là một lựa chọn tốt cho người mới học lập trình vì cú pháp đơn giản và sự dễ tiếp cận. Nó giúp người học hiểu được các khái niệm lập trình cơ bản.

Open Source và miễn phí: Python là một ngôn ngữ lập trình mã nguồn mở và miễn phí, có nghĩa bạn có thể sử dụng nó và thậm chí thay đổi mã nguồn theo ý muốn.

Python được sử dụng rộng rãi trong nhiều lĩnh vực, từ phát triển ứng dụng desktop đến phát triển trang web, phân tích dữ liệu, và nghiên cứu trong lĩnh vực trí tuệ nhân tạo. Điều này làm cho Python trở thành một công cụ quan trọng trong ngành công nghiệp phần mềm và khoa học máy tính [3].

## Các thư viện – Framework hổ trợ

**OpenCV (Open Source Computer Vision Library):**

OpenCV là một thư viện mã nguồn mở mạnh mẽ cho xử lý hình ảnh và thị giác máy tính. Trong code của bạn, bạn đã sử dụng OpenCV để thực hiện việc phát hiện và nhận diện khuôn mặt từ hình ảnh webcam.

**Tkinter:**

Tkinter là thư viện chuẩn của Python cho việc tạo giao diện người dùng đồ họa (GUI). Bạn đã sử dụng Tkinter để tạo cửa sổ và các thành phần giao diện người dùng, chẳng hạn như nút, nhãn và ô văn bản.

**Shutil:**

Thư viện shutil được sử dụng để thực hiện các thao tác liên quan đến thư mục và tệp tin, ví dụ như di chuyển các tệp ảnh từ thư mục này sang thư mục khác.

**Csv:**

Thư viện csv cho phép bạn đọc và ghi dữ liệu vào các tệp CSV, trong trường hợp này, bạn đã sử dụng nó để lưu thông tin sinh viên.

**Numpy:**

Numpy là một thư viện Python phổ biến để làm việc với mảng và ma trận số học. Trong code của bạn, bạn đã sử dụng nó để lưu các mảng hình ảnh khuôn mặt.

**Pil:**

PIL (hoặc Pillow) là thư viện Python cho xử lý hình ảnh. Bạn đã sử dụng nó để chuyển đổi hình ảnh từ định dạng PIL sang NumPy và lưu hình ảnh.

**Pandas:**

Pandas là một thư viện phổ biến cho làm việc với dữ liệu dạng bảng (dataframe). Trong code của bạn, bạn đã sử dụng pandas để đọc và ghi thông tin sinh viên vào một tệp CSV.

**Datetime:**

Thư viện datetime dùng để làm việc với thông tin thời gian và ngày tháng. Bạn đã sử dụng nó để lấy thời gian và ngày trong quá trình điểm danh.\

**Face\_recognition:**

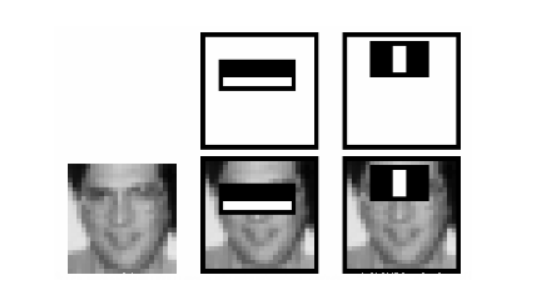
Face\_recognition là một thư viện Python phổ biến cho nhận diện khuôn mặt, sử dụng đơn giản và tích hợp chức năng phát hiện khuôn mặt trong ảnh. Nó phụ thuộc vào thư viện dlib và thường được kết hợp với OpenCV. Thư viện này được ứng dụng rộng rãi trong các lĩnh vực như xác minh người dùng và xử lý ảnh, và nhận được sự hỗ trợ tích cực từ cộng đồng Python.

# PHẦN II: CƠ SỞ LÝ THUYẾT

## Phát hiện khuôn mặt bằng mô hình Haar Cascade Classifier:

Mô hình này được sử dụng để phát hiện khuôn mặt trong hình ảnh. Nó dựa trên việc sử dụng bộ lọc Haar để xác định các đặc trưng của khuôn mặt như mắt, miệng, mũi, v.v. Sau đó, nó kiểm tra xem các đặc trưng này có tồn tại trong hình ảnh hay không để xác định khuôn mặt. Bạn có thể tải mô hình Haar Cascade Classifier cho việc nhận diện khuôn mặt từ các nguồn như OpenCV.

Trong hình minh họa bên dưới, đặc điểm hình chữ nhật đầu tiên đang tính toán cho sự khác biệt về cường độ giữa vùng mắt và vùng má trên khuôn mặt. Và đặc điểm hình chữ nhật thứ hai là đo sự chênh lệch về cường độ giữa hai vùng mắt và sống mũi. Bộ lọc Haar chỉ có thể nhìn cụ thể vào một vùng trong cửa sổ để nhận diện.



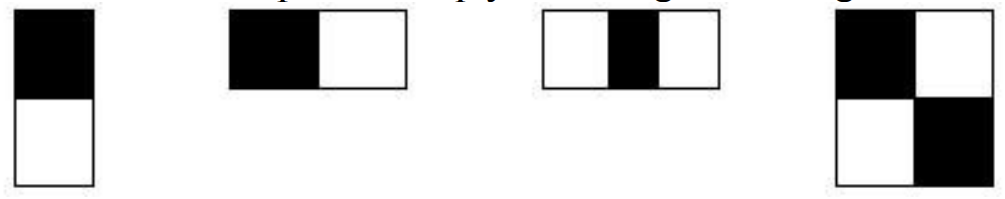
**Hình 2.1:** Một số đặc điểm haar [4].

Vì vậy, sử dụng các đặc điểm hình chữ nhật này trên một hình ảnh, chúng ta có thể tạo ra hàng nghìn điểm đặc trưng cho hình ảnh. Tuy nhiên, việc tính toán tổng số pixel trong các vùng trắng và đen trong toàn bộ ảnh có thể là một hoạt động tốn kém, đặc biệt là đối với các ảnh lớn.

Các tác giả cũng đề xuất một phương pháp được gọi là ảnh tích phân có thể đạt được tính toán tương tự bằng cách thực hiện các phép toán chỉ trên bốn điểm ảnh. Có thể sử dụng hàng nghìn hình ảnh được gắn nhãn để chuyển đổi chúng thành bản đồ tính năng tầng HAAR và đào tạo nó bằng cách sử dụng mô hình học máy để tạo bộ phân loại nhận diện khuôn mặt.

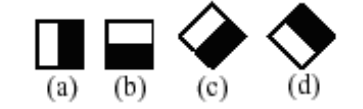
**Các đặc trưng Haar-like**

Mỗi đặc trưng Haar-like là một miền hình chữ nhật được chia thành 2,3 hoặc 4 hình chữ nhật nhỏ phân biệt quy ước bằng màu trắng và đen.

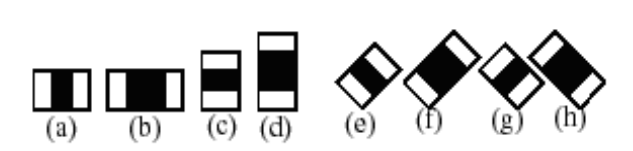
****

Như hình vẽ trên là 3 loại đặc trưng Haar-like cơ bản gồm 2,3 và 4 hình chữ nhật hợp thành. Từ những đặc trưng cơ bản mở rộng ra thành tập các đặc trưng:

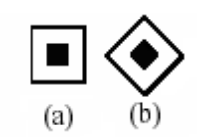
* **Đặc trưng cạnh:**

****

* **Đặc trưng đường**

****

* **Đặc trưng tâm – xung quanh**

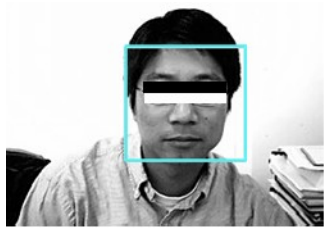
****

Giá trị của một đặc trưng Haar-like là sự khác biệt giữa tổng các giá trị xám của các pixel trong vùng “đen” với tổng các giá trị xám của các pixel trong vùng “trắng”: f(x) = Tổngvùng đen(pixel) - Tổngvùng trắng(pixel) (1) Vậy khi được đặt lên một vùng ảnh, đặc trưng Haar-like sẽ tính toán và đưa ra giá trị đặc trưng h(x) của vùng ảnh đó [4].

**Áp dụng đặc trưng Haar-like để phát hiện khuôn mặt trong ảnh.**

Như đã trình bày ở trên, để phát hiện khuôn mặt, hệ thống sẽ cho một cửa sổ con(sub-window) có kích thước cố định quét lên toàn bộ ảnh đầu vào. Như vậy sẽ có rất nhiều ảnh con tương ứng với từng cửa sổ con, các đặc trưng Haar-like sẽ đặc lên các cửa sổ con này để từ đó tính ra các giá trị của đặc trưng. Sau đó các giá trị này được bộ phân loại xác nhận xem khung hình đó có phải khuôn mặt không.

Hình dưới là một ví dụ: khung màu xanh là cửa sổ con, một đặc trưng Haar-like với kích thước và vị trí đặt như trong hình vẽ.



**Hình 2.2:** Đặc trưng haar trên khuôn mặt

## Nhận dạng khuôn mặt - Face Recognition

Face Recognition là bài toán nhận dạng và xác thực người dựa vào khuôn mặt của họ. Đối với con người thì đó là một nhiệm vụ rất đơn giản, thậm chí là ở trong những điều kiện môi trường khác nhau, tuổi tác thay đổi, đội mũ, đeo kính, … Tuy nhiên, đối với máy tính thì nó vẫn còn là một thử thách khó khăn trong vài thập kỷ qua cho đến tận ngày nay. Trong thời đại bùng nổ của trí tuệ nhân tạo, tận dụng sức mạnh của các thuật toán DL và lượng dữ liệu vô cùng lớn, chúng ta có thể tạo ra các models hiện đại, cho phép biểu diễn khuôn mặt thành các vectors đặc trưng trong không gian nhiều chiều. Để từ đó, máy tính có thể thực hiện nhận diện ra từng người riêng biệt, mà thậm chí còn vượt qua khả năng của con người trong một số trường hợp [1].

**Phân loại**

Face Recognition có thể chia thành 3 bài toán nhỏ [5]:

* Face Authentication: Hạn chế quyền truy cập của một người đến một nguồn tài nguyên nào đó.
* Face Verification: Xác nhận một người phù hợp với ID của họ.
* Face Identification: Gán chính xác tên của người.

Ba bài toán này thực ra chỉ khác nhau ở mục đích sử dụng kết quả nhận diện khuôn mặt vào việc gì, còn về bản chất vẫn là phân loại xem khuôn mặt cần nhận diện có thuộc vào nhóm nào trong bộ dữ liệu cho trước hay không?

Tất cả những bài toán này đều cần phải được giải quyết trong cả 3 trường hợp:

* Người trong ảnh
* Người trong file video
* Người thực (stream real-time từ camera)

Tuy nhiên, cũng lại xuất hiện thêm một bài toán con con nữa, đó là đôi khi chúng ta cần phân biệt đâu là người thật, đâu là người giả (người trong video hay ảnh). Vì nếu chúng ta đối xử với cả 3 trường hợp đều như nhau thì rất có thể kẻ gian sẽ lợi dụng để truy cập trái phép vào hệ thống thông qua một bức ảnh, cái mà rất dễ dàng có được.

# PHẦN III: XÂY DỰNG CHƯƠNG TRÌNH

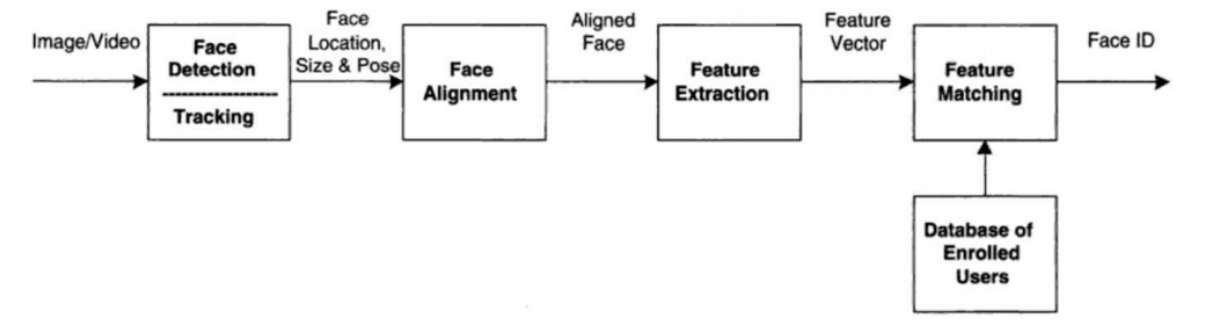
## Xây dựng mô hình Face Recognition

Face Recognition bắt buộc phải bao gồm tối thiếu 3 bước sau [5]:

* Bước 1: Face Detection - Xác định vị trí của khuôn mặt trong ảnh (hoặc video frame). Vùng này sẽ được đánh dấu bằng một hình chữ nhật bao quanh.
* Bước 2: Face Extraction (Face Embedding) - Trích xuất đặc trưng của khuôn mặt thành một vector đặc trưng trong không gian nhiều chiểu (thường là 128 chiều).
* Bước 3: Face Classification (Face Authentication - Face Verification - Face Identification).

Ngoài 3 bước trên, trong thực tế chúng ta thường bổ sung thêm một số bước để tăng độ chính xác nhận diện:

* Image Preprocessing: Xử lý giảm nhiễu, giảm mờ, giảm kích thước, chuyển sang ảnh xám, chuẩn hóa, …
* Face Aligment: Nếu ảnh khuôn mặt bị nghiêng thì căn chỉnh lại sao cho ngay ngắn.
* Kết hợp nhiều phương pháp khác nhau tại bước 3.



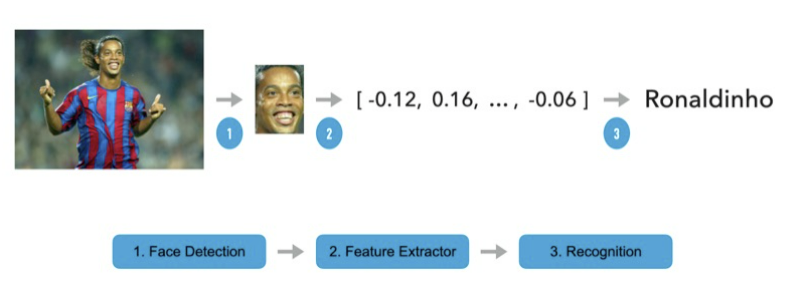
**Hình 3.1:** Sơ đồ các bước nhận dạng bằng face-recognition

Sơ đồ face-recognition là một biểu đồ mô tả các bước cơ bản của quá trình nhận dạng khuôn mặt. Quá trình này bao gồm các bước sau:

1. Khám phá: Bước này bao gồm việc xác định khuôn mặt trong hình ảnh hoặc video.
2. Định vị: Bước này bao gồm việc xác định vị trí của khuôn mặt trong hình ảnh hoặc video.
3. Tính năng: Bước này bao gồm việc trích xuất các đặc điểm của khuôn mặt, chẳng hạn như vị trí của mắt, mũi, miệng, v.v.
4. So sánh: Bước này bao gồm việc so sánh các đặc điểm của khuôn mặt được trích xuất với các đặc điểm của khuôn mặt được lưu trữ trong cơ sở dữ liệu.
5. Xác nhận: Bước này bao gồm việc xác nhận xem khuôn mặt được phát hiện có khớp với bất kỳ khuôn mặt nào được lưu trữ trong cơ sở dữ liệu hay không.

## Các bước thực hiện

**Face recognition hoạt động như thế nào?**



**Hình 3.1:** Quy trình nhận diện khuôn mặt [6].

Nhận dạng khuôn mặt là công nghệ xác định danh tính của một cá nhân dựa trên khuôn mặt của họ. Một hệ thống hay phần mềm nhận dạng khuôn mặt thường bao gồm 3 bước chính:

* Xác định vị trí khuôn mặt (Face Detection)
* Trích xuất đặc trưng ảnh (Feature Extraction)
* Nhận dạng khuôn mặt (Recognition)

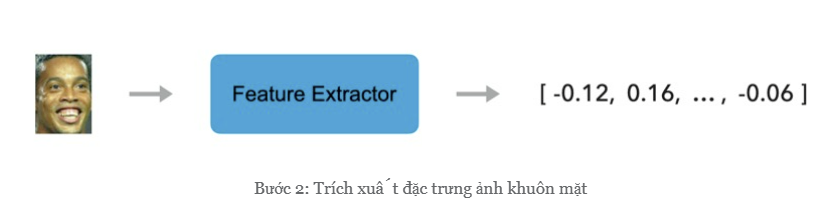
**Xác định vị trí khuôn mặt (Face Detection)**



**Hình 3.2:** Xác định khuôn mặt trong ảnh

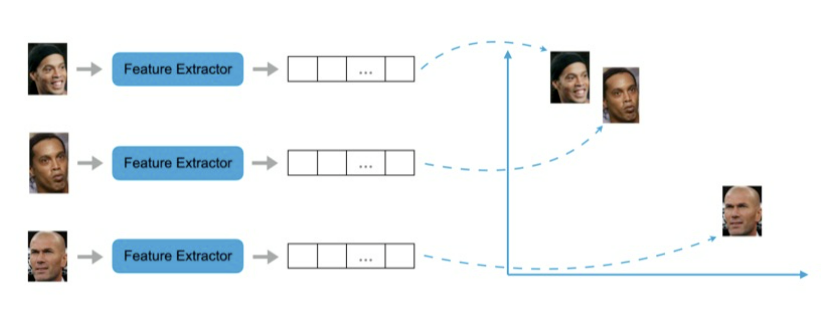
Đây là bước xác định khuôn mặt người từ những bức ảnh hoặc băng hình (images or videos). Những khuôn mặt này sẽ được “cắt” ra để dùng cho bước tiếp theo.

**Trích xuất đặc trưng ảnh (Feature Extraction)**

****

**Hình 3.3:** Trích xuất đặc trưng ảnh

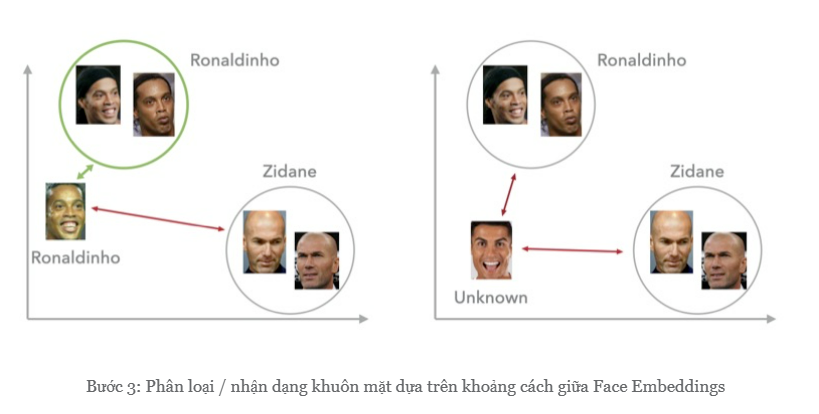
Ở bước này, chúng ta sẽ cần một mô hình để trích xuất đặc trưng từ ảnh khuôn mặt (Feature Extractor). Những đặc trưng được trích xuất từ mô hình này còn được gọi là Face Embeddings hay Face Encodings, chúng ta có thể hiểu là cách biểu diễn khác của ảnh khuôn mặt giúp cho việc nhận dạng khuôn mặt dễ dàng hơn (thay vì những điểm ảnh đơn thuần). Face Embeddings có một đặc tính rất thú vị: những khuôn mặt giống nhau sẽ có Face Embeddings “gần” nhau hơn (dựa trên khoảng cách Euclid) [6].



**Hình 3.4:** Các ảnh có đặc trưng gần giống nhau

Điều này đạt được khi mô hình trích xuất đặc trưng được thiết kế riêng cho việc phân biệt / nhận dạng khuôn mặt.

**Nhận dạng khuôn mặt (Recognition)**

****

**Hình 3.5:** Phân loại dựa vào khoảng cách embeddings

Với đặc tính của Face Embeddings đề cập ở trên, một phương pháp đơn giản để nhận dạng khuôn mặt là dựa trên khoảng cách giữa các Face Embeddings với nhau. Hãy tưởng tượng mỗi khuôn mặt có thể được biểu diễn bằng một điểm toạ độ trên trục xy, nếu khoảng cách giữa 2 điểm càng gần nhau, khả năng 2 khuôn mặt thuộc về cùng 1 người càng cao. Ngược lại, nếu khoảng cách này quá lớn (vượt qua một ngưỡng nhất định), chúng ta có thể “tự tin” rằng 2 khuôn mặt thuộc về 2 người khác nhau [6].

## Cấu hình ESP32-CAM

Đoạn mã tạo ra một máy ảnh IP có khả năng cập nhật firmware từ xa và có thể chụp ảnh ở các độ phân giải khác nhau dựa trên yêu cầu từ các URL cụ thể. Mã nguồn này giúp tạo ra một giải pháp đơn giản cho ứng dụng quan sát hoặc giám sát trong các dự án IoT.

#include <dummy.h>

#include <WebServer.h>

#include <WiFi.h>

#include <esp32cam.h>

const char\* WIFI\_SSID = "hi";

const char\* WIFI\_PASS = "88888888";

WebServer server(80);

static auto loRes = esp32cam::Resolution::find(320, 240);

static auto midRes = esp32cam::Resolution::find(350, 530);

static auto hiRes = esp32cam::Resolution::find(800, 600);

void serveJpg()

{

  auto frame = esp32cam::capture();

  if (frame == nullptr) {

    Serial.println("CAPTURE FAIL");

    server.send(503, "", "");

    return;

  }

  Serial.printf("CAPTURE OK %dx%d %db\n", frame->getWidth(), frame->getHeight(),

                static\_cast<int>(frame->size()));

  server.setContentLength(frame->size());

  server.send(200, "image/jpeg");

  WiFiClient client = server.client();

  frame->writeTo(client);

}

void handleJpgLo()

{

  if (!esp32cam::Camera.changeResolution(loRes)) {

    Serial.println("SET-LO-RES FAIL");

  }

  serveJpg();

}

void handleJpgHi()

{

  if (!esp32cam::Camera.changeResolution(hiRes)) {

    Serial.println("SET-HI-RES FAIL");

  }

  serveJpg();

}

void handleJpgMid()

{

  if (!esp32cam::Camera.changeResolution(midRes)) {

    Serial.println("SET-MID-RES FAIL");

  }

  serveJpg();

}

void  setup(){

  Serial.begin(115200);

  Serial.println();

  {

    using namespace esp32cam;

    Config cfg;

    cfg.setPins(pins::AiThinker);

    cfg.setResolution(hiRes);

    cfg.setBufferCount(2);

    cfg.setJpeg(80);

    bool ok = Camera.begin(cfg);

    Serial.println(ok ? "CAMERA OK " : "CAMERA FAIL");

  }

  WiFi.persistent(false);

  WiFi.mode(WIFI\_STA);

  WiFi.begin(WIFI\_SSID, WIFI\_PASS);

  while (WiFi.status() != WL\_CONNECTED) {

    delay(500);

  }

  Serial.print("http://");

  Serial.println(WiFi.localIP());

  Serial.println("  /cam-lo.jpg");

  Serial.println("  /cam-hi.jpg");

  Serial.println("  /cam-mid.jpg");

  server.on("/cam-lo.jpg", handleJpgLo);

  server.on("/cam-hi.jpg", handleJpgHi);

  server.on("/cam-mid.jpg", handleJpgMid);

  server.begin();

}

void loop()

{

  server.handleClient();

}

# PHẦN IV: KẾT QUẢ

**Giao diện:**



**Hình 4.1:** Giao diện

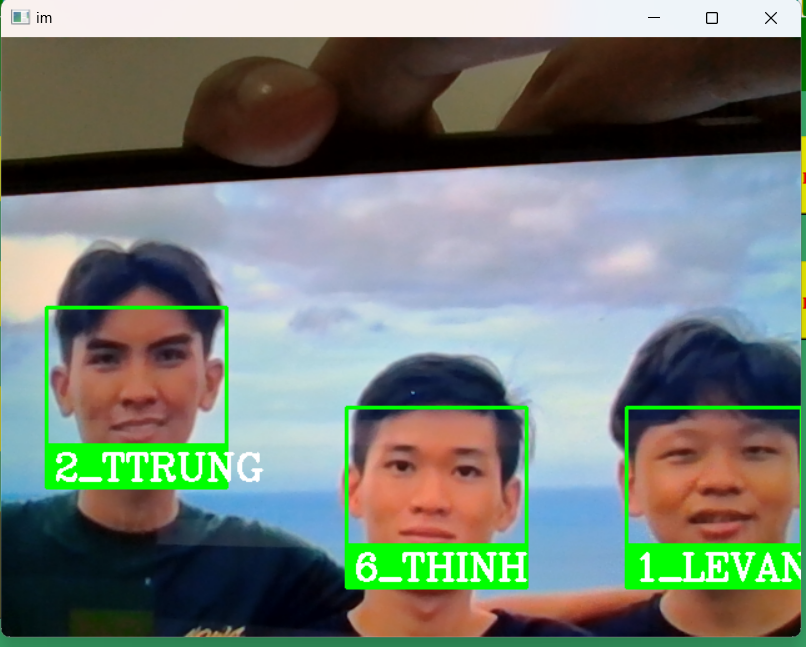
**Nhập ID và Tên để lấy ảnh mẫu:**



**Hình 4.2:** Lấy ảnh mẫu

**Sau khi lấy ảnh mẫu xong, ấn vào Train ảnh:**

**Trai ảnh xong có thể nhận diện khuôn mặt thông qua camera:**



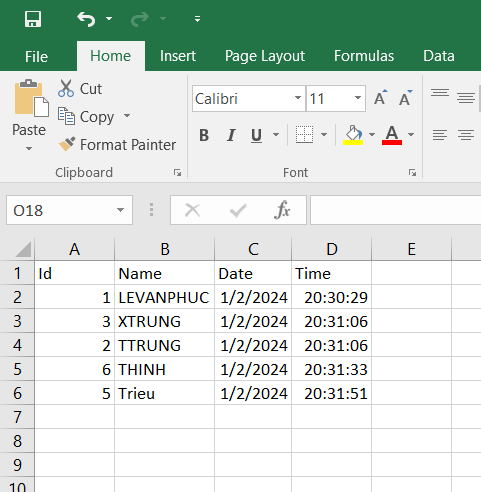
**Hình 4.3**: Camera nhận diện các khuôn mặt

**Khi đã nhận diện xong các khuôn mặt sẽ thông báo số sinh viên hiện đang có mặt:**



**Hình 4.4:** Thông báo sỉ số

**Có thể xem lại tất cả sinh viên có mặt trong file excel:**



**Hình 4.5:** Lưu thông tin sinh viên có mặt

# PHẦN V: TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN

## Tổng kết:

"Nhận diện khuôn mặt dựa vào học máy" đại diện cho một bước tiến quan trọng trong sự hòa nhập giữa công nghệ và quản lý sự hiện diện. Trong quá trình nghiên cứu và phát triển, chúng ta đã chứng kiến sức mạnh của các phương pháp nhận diện khuôn mặt, đặc biệt là mô hình Haar Cascade và LBPHFaceRecognizer của OpenCV, trong việc tạo ra một hệ thống đơn giản và hiệu quả.

Mô hình Haar Cascade đã đem lại sự linh hoạt và tốc độ trong quá trình nhận diện khuôn mặt, đặc biệt là trong điều kiện ổn định với ánh sáng đủ. Nó đã mở ra cánh cửa cho nhiều ứng dụng thực tế, từ quản lý sự hiện diện trong doanh nghiệp, giáo dục đến tổ chức sự kiện và nhiều ngữ cảnh khác.

Tuy nhiên, để đưa đề tài này lên một tầm cao mới, chúng ta cần nhìn xa hơn và tận dụng những tiềm năng phát triển. Hướng phát triển đề xuất chú trọng vào việc tích hợp công nghệ tiên tiến như mạng nơ-ron tích chập để cải thiện độ chính xác, đồng thời xem xét khả năng ứng dụng di động để mang lại sự linh hoạt và tiện ích trong thực tế.

Nhìn chung, "Nhận diện khuôn mặt dựa vào học máy" không chỉ là một công cụ quản lý sự hiện diện mà còn là một bước tiến quan trọng trong việc kết nối công nghệ và con người để tạo ra một giải pháp thực sự hiệu quả và linh hoạt trong thế giới hiện đại.

## Hướng phát triển:

* **Tích hợp Công Nghệ Tiên Tiến:**

Nghiên cứu và tích hợp các công nghệ nhận diện khuôn mặt tiên tiến như mạng nơ-ron tích chập (CNN) để cải thiện độ chính xác và độ linh hoạt của hệ thống.

* **Phát triển Ứng Dụng Di Động:**

Tích hợp hệ thống vào ứng dụng di động để tạo ra một giải pháp di động cho việc điểm danh và quản lý sự hiện diện, đặc biệt là trong các tổ chức và sự kiện lớn.

* **Tăng Cường Bảo Mật và Quyền Riêng Tư:**

Nâng cao lớp bảo mật để đảm bảo an toàn cho dữ liệu nhận diện khuôn mặt và tuân thủ nghiêm ngặt các chuẩn quyền riêng tư.

* **Thử Nghiệm và Tối Ưu Hóa Hiệu Suất:**

Tiến hành thử nghiệm rộng rãi trên nhiều tình huống và điều kiện ánh sáng khác nhau để đảm bảo rằng hệ thống hoạt động ổn định và chính xác trong mọi tình huống.

* **Phát triển Giao Diện Người Dùng Thân Thiện:**

Thiết kế giao diện người dùng thân thiện và dễ sử dụng để tối ưu hóa trải nghiệm người dùng khi sử dụng hệ thống.

* **Liên Kết với Cơ Sở Dữ Liệu và Hệ Thống Khác:**

Kết nối hệ thống với cơ sở dữ liệu và hệ thống quản lý khác để tăng khả năng tích hợp và tương tác với các phần mềm và dịch vụ khác.

Tổng cộng, những hướng phát triển này sẽ giúp đưa đề tài "Điểm danh bằng nhận diện khuôn mặt" lên một tầm cao mới, đồng thời mang lại giải pháp hiệu quả và đa chiều cho việc quản lý sự hiện diện trong nhiều bối cảnh khác nhau.

# PHỤ LỤC

**Một số đoạn code quan trọng trong bài:**

***Kiểm tra ID đã được sử dụng chưa:***

# Kiểm tra ID ảnh có trùng hay không  
if (name.isalpha() and is\_number(Id)):  
 res = "Id đã được sử dụng"  
 message.configure(text=res)  
# Kiểm tra ID ảnh có trùng hay không

***Code chụp ảnh:***

# bắt đầu chụp ảnh  
while (True):  
 # Camera máy tính  
 # ret, img = cam.read()  
  
 # Camera esp32  
 img\_resp = urllib.request.urlopen(url)  
 imgnp = np.array(bytearray(img\_resp.read()), dtype=np.uint8)  
 img = cv2.imdecode(imgnp, -1)  
  
 # Tìm khuôn mặt trong cam  
 faces = detector.detectMultiScale(img, 1.3, 5)  
 cv2.imshow('frame', img)  
 # Chụp ảnh và lưu vào thư mục TrainingImage  
 for (x, y, w, h) in faces:  
 cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)  
 # Đổi kích thước ảnh về 300x330  
 resized\_face = cv2.resize(img[y:y + h, x:x + w], (300, 330))  
 # incrementing sample number  
 sampleNum = sampleNum + 1  
 # TrainingImageTemp dùng để train và đưa tất cả ảnh vừa train vào cùng 1 thư mục  
 cv2.imwrite("TrainingImageTemp\\" + Id + "\_" + name + '\_' + str(sampleNum) + ".jpg",  
 resized\_face)  
 # display the frame  
 cv2.imshow('frame', img)  
  
 # dừng khi ấn "q" hoặc số ảnh chụp >100  
 if cv2.waitKey(100) & 0xFF == ord('q'):  
 break  
 # break if the sample number is morethan 100  
 elif sampleNum > 5:  
 break

***Train ảnh:***

#train các ảnh đã chụp  
def TrainImages():  
 # Chạy vòng for đưa hết ảnh ở train tạm sang DoneTrain  
 for imageDone in os.listdir('TrainingImageTemp\\'):  
 destination\_folder = os.path.join(CurDir, "TrainingImage", str(txt.get()) + "\_" + str(txt2.get()))  
 # Tạo thư mục đích nếu nó không tồn tại  
 os.makedirs(destination\_folder, exist\_ok=True)  
 destination\_path = os.path.join(destination\_folder, imageDone)  
 MoveFile(os.path.join(CurDir, "TrainingImageTemp", imageDone), destination\_path)  
 res = "Image Trained"#+",".join(str(f) for f in Id)  
 message.configure(text= res)

***Đếm tổng số lượng sinh viên:***

# Đếm tổng số lượng sinh viên  
def count\_sv(directory):  
 if not os.path.exists(directory):  
 return 0  
 SiSoSinhVien = 0  
 for root, dirs, files in os.walk(directory):  
 SiSoSinhVien += len(dirs)  
 return SiSoSinhVien  
# Đếm tổng số lượng sinh viên

***Nhận diện khuôn mặt:***

# Thực hiện vòng lặp vô hạn để liên tục nhận diện khuôn mặt từ máy ảnh.  
while True:  
 # Camera máy tính  
 ret, img = cam.read()  
  
 # Camera esp32  
 # img\_resp = urllib.request.urlopen(url)  
 # imgnp = np.array(bytearray(img\_resp.read()), dtype=np.uint8)  
 # img = cv2.imdecode(imgnp, -1)  
  
 imgS = cv2.resize(img, (0, 0), None, 0.25, 0.25)  
 imgS = cv2.cvtColor(imgS, cv2.COLOR\_BGR2RGB)  
  
 facesCurFrame = face\_recognition.face\_locations(imgS)  
 encodesCurFrame = face\_recognition.face\_encodings(imgS, facesCurFrame)  
  
 for encodeFace, faceLoc in zip(encodesCurFrame, facesCurFrame):  
 print("Face Detected")  
 matches = face\_recognition.compare\_faces(encodeListKnown, encodeFace)  
 faceDis = face\_recognition.face\_distance(encodeListKnown, encodeFace)  
 # print(faceDis)  
 matchIndex = np.argmin(faceDis)  
  
 if matches[matchIndex]:  
 name = classNames[matchIndex].upper()  
  
 # Tách tên file và lấy phần cần thiết  
 parts = name.split("\_")  
 if len(parts) >= 2:  
 name\_new = "\_".join(parts[:2])  
 else:  
 name\_new = name  
  
 ts = time.time()  
 date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')  
 timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')  
  
 Id = parts[0]  
 Name = parts[1]  
  
 y1, x2, y2, x1 = faceLoc  
 y1, x2, y2, x1 = y1 \* 4, x2 \* 4, y2 \* 4, x1 \* 4  
 cv2.rectangle(img, (x1, y1), (x2, y2), (0, 255, 0), 2)  
 cv2.rectangle(img, (x1, y2 - 35), (x2, y2), (0, 255, 0), cv2.FILLED)  
 cv2.putText(img, name\_new, (x1 + 6, y2 - 6), cv2.FONT\_HERSHEY\_COMPLEX, 1, (255, 255, 255), 2)  
  
 attendance.loc[len(attendance)] = [Id, Name, date, timeStamp]  
  
 attendance = attendance.drop\_duplicates(subset=['Id'], keep='first')  
 cv2.imshow('im', img)  
 if (cv2.waitKey(1) == ord('q')):  
 break

***Đếm số lượng sinh viên có mặt:***

# Đếm số lượng dòng trong file csv -> số sinh viên có mặt  
SV\_Attendance = attendance.shape[0]  
print(f"Số sinh viên có mặt: {SV\_Attendance}")  
  
# Chuyển sang kiểu chuỗi đưa vào Message3  
SoSvThamGia = str(SV\_Attendance)  
  
# Đếm số lượng sv  
SiSoSinhVien = count\_sv("TrainingImage\\")  
print(f"Tổng số lượng sinh viên: {SiSoSinhVien}")  
  
# Chuyển sang kiểu chuổi đưa vào Message3  
TongsisoSV = str(SiSoSinhVien)  
  
res1 = "Sỉ số: " + SoSvThamGia + "/" + TongsisoSV  
message3.configure(text=res1)  
# Đếm số lượng sv  
  
print(attendance)  
#print(attendance)  
res=attendance  
message2.configure(text= res)

# References

|  |  |
| --- | --- |
| [1] | Tran Van Huy, "Face Recognition là gì?," [Online]. Available: https://huytranvan2010.github.io/Face-Recognition/#:~:text=Qu%C3%A1%20tr%C3%ACnh%20nh%E1%BA%ADn%20di%E1%BB%87n%20khu%C3%B4n%20m%E1%BA%B7t%20bao%20g%E1%BB%93m,tr%C3%AAn%20Siamese%20Network%2C%20Local%20Bianry%20Patterns%20%28LBPs%29%2C%20Eigenfaces%E2%80%A. |
| [2] | luuvachiase, "ESP32-CAMERA: Cài đặt môi trường Arduino IDE và nạp chương trình," [Online]. Available: https://luuvachiase.net/index.php/2019/12/27/esp32-camera-cai-dat-moi-truong-arduino-ide-va-nap-chuong-trinh/#:~:text=Esp32-Cam%20l%C3%A0%20m%E1%BB%99t%20module%20d%E1%BB%B1a%20tr%C3%AAn%20SoC%20ESP32,%C3%A1n%20IoT%20v%E1%BB%81%20gi%C3%A1m%20s%C3%A1t%20v%. |
| [3] | glints, "Python Là Gì? Tất Tần Tật Về Ngôn Ngữ Lập Trình Python," [Online]. Available: https://glints.com/vn/blog/ngon-ngu-lap-trinh-python-la-gi/. |
| [4] | Ngoc N Tran, "Haar Cascade là gì? Luận về một kỹ thuật chuyên dùng để nhận biết các khuôn mặt trong ảnh.," [Online]. Available: https://viblo.asia/p/haar-cascade-la-gi-luan-ve-mot-ky-thuat-chuyen-dung-de-nhan-biet-cac-khuon-mat-trong-anh-E375zamdlGW. |
| [5] | SuNT, "Bài toán Face Recognition," [Online]. Available: https://tiensu.github.io/blog/52\_face\_recognition/#:~:text=B%C3%A0i%20to%C3%A1n%20Face%20Recognition%20b%E1%BA%AFt%20bu%E1%BB%99c%20ph%E1%BA%A3i%20bao,Authentication%20-%20Face%20Verification%20-%20Face%20Identification%20%29.. |
| [6] | Duy Hưng, "Tìm hiểu công nghệ nhận dạng khuôn mặt - Face Recognition," [Online]. Available: https://www.pandaml.com/nhan-dang-khuon-mat/. |