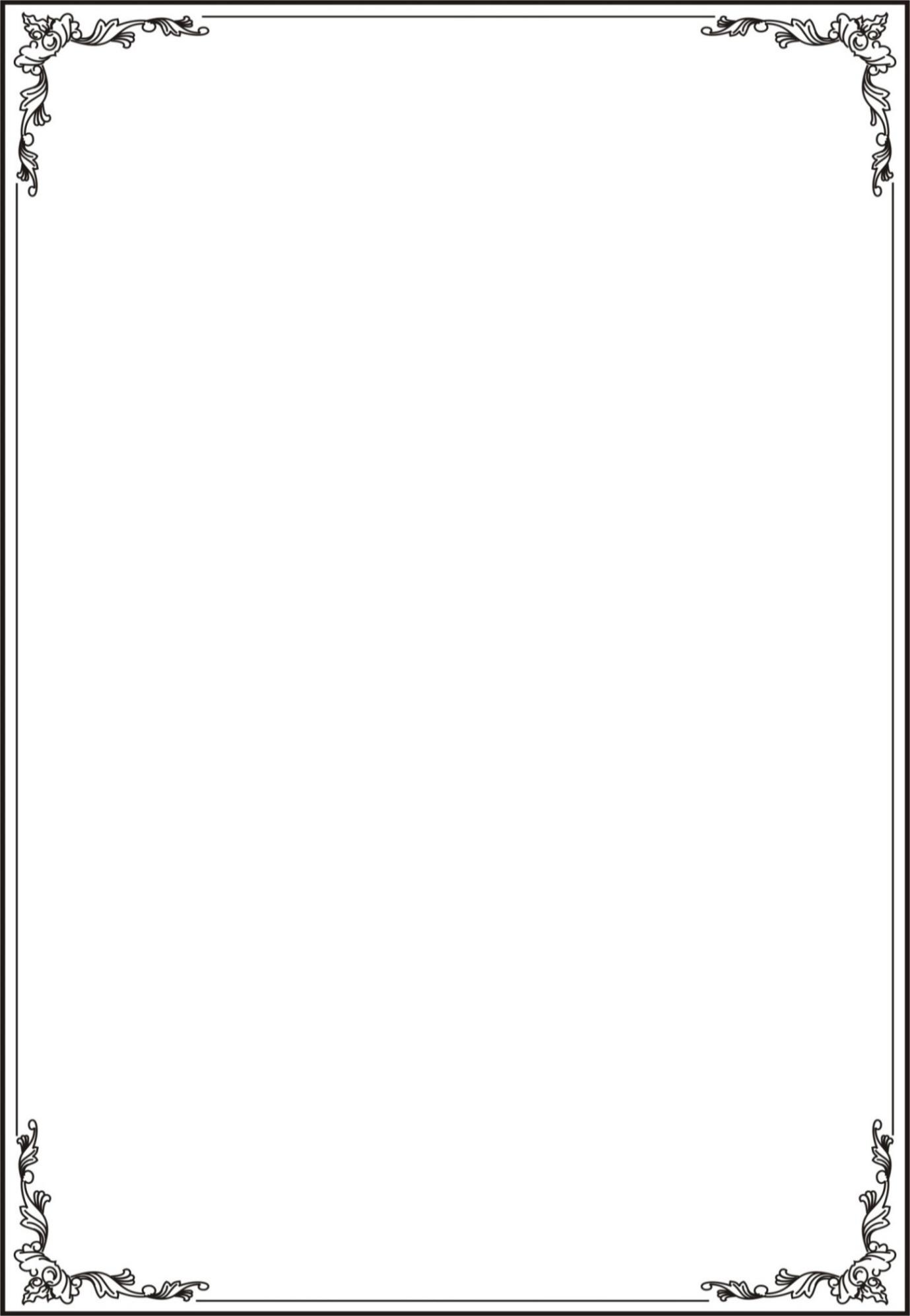
**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT HÀN**



**ĐỒ ÁN CƠ SỞ 5**

**XÂY DỰNG HỆ THỐNG NHẬN DIỆN VÀ ĐIỂM DANH KHUÔN MẶT BẰNG PYTHON**

**Sinh viên thực hiện:**

Nguyễn Ngọc Quang 18IT5

Nguyễn Trọng Tâm 18IT5

**Giáo viên hướng dẫn:** TS.Nguyễn Sỹ Thìn

***Đà nẵng, tháng 5 năm 2021***

**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT HÀN**



**ĐỒ ÁN CƠ SỞ 5**

**XÂY DỰNG HỆ THỐNG NHẬN DIỆN VÀ ĐIỂM DANH KHUÔN MẶT BẰNG PYTHON**

**Sinh viên thực hiện:**

Nguyễn Ngọc Quang 18IT5

Nguyễn Trọng Tâm 18IT5

**Giáo viên hướng dẫn:** TS.Nguyễn Sỹ Thìn

***Đà nẵng, tháng 5 năm 2021***

**LỜI CẢM ƠN**

Em xin được chân thành cảm ơn sự giúp đỡ tận tình của thầy giáo Đỗ Công Đức đã cung cấp cho chúng em thêm những kiến thức trình duyệt web cũng như định hướng cho chúng em những phương pháp lập trình và cung cấp tài liệu để chúng em có thể hoàn thành tốt đồ án này.

Chúng em cũng xin trân trọng cảm ơn các thầy cô giáo giảng viên trong hội đồng sư phạm nhà Trường Đại Học Công Nghệ Thông Tin và Truyền Thông Việt - Hàn đã giúp chúng em hiểu rõ hơn để có thể tạo ra một giao diện đẹp và có nhiều tính năng và đã giúp đỡ chúng em rất nhiều trong thời gian học tập tại trường.

Chúng em cũng xin gửi lời cảm ơn đến gia đình, bạn bè luôn động viên giúp đỡ chúng em trong suốt thời gian học tập và nghiên cứu, đóng góp những kinh nghiệm quý báu trong thời gian thực hiện đề tài này.

Kính chúc thầy cô mạnh khỏe, công tác tốt, tiếp tục giảng dạy và đào tạo thế hệ trẻ thành công. Một lần nữa chúng em trân trọng cảm ơn

**MỤC LỤC**

[Chương 1 TỔNG QUAN 2](#_Toc73323767)

[1.1 Tổng quan về nhận diện khuôn mặt 2](#_Toc73323768)

[1.2 Phân tích bài toán 2](#_Toc73323769)

[1.3 Tổng quan về hệ thống 4](#_Toc73323770)

[1.4 Các công cụ 4](#_Toc73323771)

[1.4.1 Python 5](#_Toc73323772)

[1.4.2 Thư viện OpenCV 6](#_Toc73323773)

[1.4.3 Visual Studio Code 7](#_Toc73323774)

[1.4.4 SQLite 7](#_Toc73323775)

[1.5 Cấu trúc báo cáo 8](#_Toc73323776)

[Chương 2 KỸ THUẬT NHẬN DẠNG KHUÔN MẶT 9](#_Toc73323777)

[2.1 Phát hiện khuôn mặt 9](#_Toc73323778)

[2.1.1 Vấn đề 9](#_Toc73323779)

[2.1.2 Tìm khuôn mặt 10](#_Toc73323780)

[2.1.3 Phương pháp nhận diện khuôn mặt của Violas & John 11](#_Toc73323781)

[2.2 Tiền xử lý ảnh 17](#_Toc73323782)

[2.3 Trích rút đặc trưng 17](#_Toc73323783)

[2.4 Dữ liệu hệ thống nhận dạng 17](#_Toc73323784)

[Chương 3 PHÂN TÍCH THIẾT KẾ VÀ TRIỂN KHAI THỰC HIỆN 19](#_Toc73323785)

[3.1 Phân tích thiết kế hệ thống 19](#_Toc73323786)

[3.1.1 Sơ đồ Use-case 19](#_Toc73323787)

[3.1.2 Mô hình sequence diagram 19](#_Toc73323788)

[3.2 Cài đặt môi trường 20](#_Toc73323789)

[3.2.1 Python 20](#_Toc73323790)

[3.2.2 Visual studio code 21](#_Toc73323791)

[3.2.3 Cài đặt OpenCV 23](#_Toc73323792)

[3.3 Xây dựng phần mềm 23](#_Toc73323793)

[Chương 4 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 28](#_Toc73323794)

[4.1 Đánh giá kết quả 28](#_Toc73323795)

[4.1.1 Kết quả đạt được 28](#_Toc73323796)

[4.1.2 Hạn chế 28](#_Toc73323797)

[4.2 Hướng phát triển 28](#_Toc73323798)

**DANH MỤC HÌNH**

Trang

[Hình 1.1 Hệ thống nhận dạng khuôn mặt 3](#_Toc73323830)

[Hình 1.2 Ngôn ngữ Python 5](#_Toc73323831)

[Hình 2.1 Hệ thống nhận diện khuôn mặt 9](#_Toc73323832)

[Hình 2.2 Đặc trưng Haar-Like 12](#_Toc73323833)

[Hình 2.3 Bốn đặc trưng cơ bản 12](#_Toc73323834)

[Hình 2.4 Công thức tính trong Haar-Like 13](#_Toc73323835)

[Hình 2.5 Integral Image 13](#_Toc73323836)

[Hình 2.6 Tính tổng mức xám vùng D 14](#_Toc73323837)

[Hình 2.7 AdaBoost 15](#_Toc73323838)

[Hình 2.8 Kết hợp bộ phân loại yếu thành mạnh 16](#_Toc73323839)

[Hình 2.9 Sơ đồ phát hiện khuôn mặt 16](#_Toc73323840)

[Hình 3.1 Use case Camera và Chương trình 19](#_Toc73323841)

[Hình 3.2 Mô hình chức năng nhận diện camera 19](#_Toc73323842)

[Hình 3.3 Chức năng thông báo, chỉnh sửa danh sách điểm danh 20](#_Toc73323843)

[Hình 3.4 Cài đặt Python 21](#_Toc73323844)

[Hình 3.5 Tải Visual Studio Code 22](#_Toc73323845)

[Hình 3.6 Cài đặt Visual Studio Code 22](#_Toc73323846)

[Hình 3.7 Thêm Python vào phần mềm 23](#_Toc73323847)

[Hình 3.8 Giao diện đăng nhập 24](#_Toc73323848)

[Hình 3.9 Bảng điều khiển 24](#_Toc73323849)

[Hình 3.10 Quản lý thông tin sinh viên 25](#_Toc73323850)

[Hình 3.11 Xây dựng mô hình nhận diện 26](#_Toc73323851)

[Hình 3.12 Điểm danh các môn học 26](#_Toc73323852)

[Hình 3.13 Thông tin sinh viên khi được nhận dạng 27](#_Toc73323853)

**MỞ ĐẦU**

Khuôn mặt đóng vai trò quan trọng trong quá trình giao tiếp giữa người với người, và cũng mang một lượng thông tin giàu có, chẳng hạn có thể xác định giới tính, tuổi tác, trạng thái cảm xúc của người đó, ... hơn nữa khảo sát chuyển động của các đường nét trên khuôn mặt có thể biết được người đó đang muốn nói gì. Trong hệ thống nhận dạng người thì quá trình nhận dạng khuôn mặt được đánh giá là bước khó khăn và quan trọng nhất so với các bước còn lại của hệ thống. Do đó, nhận dạng khuôn mặt là điều quan trọng và cần thiết. Nhận dạng khuôn mặt người là một công nghệ được ứng dụng rộng rãi trong đời sống hằng ngày của con người như các hệ thống giám sát, quản lý vào ra, tìm kiếm thông tin người nổi tiếng, ... có rất nhiều phương pháp nhận dạng khuôn mặt để nâng cao hiệu suất tuy nhiên dù ít hay nhiều những phương pháp này đang vấp phải những thử thách về độ sáng, hướng nghiên, kích thước ảnh, hay ảnh hưởng của tham số môi trường. Cùng với sự phát triển của xã hội, vấn đề an ninh, bảo mật đang được yêu cầu khắt khe tại mọi quốc gia trên thế giới. Các hệ thống nhận dạng con người, đồ vật ... được ra đời và phát triển với độ tin cậy ngày càng cao. Với cách tiếp cận đối tượng nhận dạng theo phương pháp này, chúng ta có thể thu nhập được nhiều thông tin từ đối tượng hơn, mà không cần tác động nhiều đến đối tượng cũng vẫn đảm bảo tính chính xác, an toàn, thuận tiện. Trong phạm vi bài báo cáo này chúng em xin được trình bài quá trình thực hiện nhận diện khuôn mặt qua thư viện opencv. Đề tài chúng em chọn là: " XÂY DỰNG HỆ THỐNG NHẬN DIỆN VÀ ĐIỂM DANH KHUÔN MẶT BẰNG PYTHON."

# TỔNG QUAN

## Tổng quan về nhận diện khuôn mặt

Hơn một thập kỷ qua rất nhiều công trình nghiên cứu về bài toán xác định khuôn mặt người từ ảnh đen trắng, xám đến ảnh màu như ngày hôm nay. Các nhà nghiên cứu di từ bài toán đơn giản, mỗi ảnh chỉ có một mặt người nhìn thẳng vào thiết bị thu hình và đầu ở tư thế thẳng đứng trong ảnh đen trắng. Cho đến ngày hôm nay bài toán mở rộng cho ảnh màu, có nhiều khuôn mặt trong cùng một ảnh, có nhiều tư thế thay đổi trong ảnh. Không những thế mà còn mở rộng cả phạm vi từ môi trường xung quanh khá đơn giản cho đến môi trường xung quanh rất phức tạp nhằm đáp ứng nhu cầu của con người. Trong những năm gần đây các ứng dụng về trí tuệ nhân tạo ngày càng phát triển và được đánh giá cao. Một lĩnh vực đang được quan tâm của trí tuệ nhân tạo nhằm tạo ra các ứng dụng thông minh, có tính người đó là nhận dạng. Trong đề tài này em chọn đối tượng là khuôn mặt. Và em chọn đề tài: "XÂY DỰNG HỆ THỐNG NHẬN DIỆN VÀ ĐIỂM DANH KHUÔN MẶT BẰNG PYTHON."

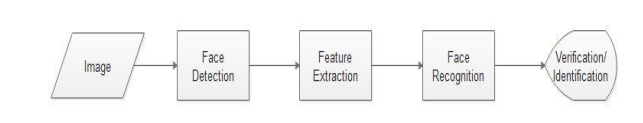
## Phân tích bài toán

Nhận dạng khuôn mặt (Face Recognition) là một phương pháp sinh trắc để xác định hoặc xác minh một cá nhân nào đó bằng cách so sánh dữ liệu hình ảnh chụp trực tiếp hoặc hình ảnh kỹ thuật số với bản ghi được lưu trữ cho người đó. Nó được xem là một lĩnh vực nghiên cứu của ngành Biometrics (tương tự như nhận dạng vân tay – Fingerprint Recognition, hay nhận dạng mống mắt – Iris Recognition). Xét về nguyên tắc chung, nhận dạng khuôn mặt có sự tương đồng rất lớn với nhận dạng vân tay và nhận dạng mống mắt, tuy nhiên sự khác biệt nằm ở bước trích chọn đặt trưng (feature extraction) của mỗi lĩnh vực. Trong khi nhận dạng vân tay và mống mắt đã đạt tới độ chín, tức là có thể áp dụng trên thực tế một cách rộng rãi thì nhận dạng khuôn mặt người vẫn còn nhiều thách thức và vẫn là một lĩnh vực nghiên cứu thú vị với nhiều người.So với nhận dạng vân tay và mống mắt, nhận dạng khuôn mặt có nguồn dữ liệu phong phú hơn (chúng ta có thể nhìn thấy mặt người ở bất cứ tấm ảnh, video clip nào liên quan tới con người trên mạng) và ít đòi hỏi sự tương tác có kiểm soát hơn (để thực hiện nhận dạng vân tay hay mống mắt, dữ liệu input lấy từ con người đòi hỏi có sự hợp tác trong môi trường có kiểm soát).

Các hệ thống nhận dạng khuôn mặt thường được sử dụng cho các mục đích an ninh như kiểm soát an ninh tại tòa nhà, sân bay, máy ATM, tra cứu thông tin của tội phạm, phát hiện tội phạm ở nơi công cộng, ...và ngày càng được ứng dụng rộng rãi trong cuộc sống

Bên cạnh những thành công đã được ghi nhận thì nhận dạng khuôn mặt cũng còn gặp nhiều khó khăn như về độ sáng, hướng nghiêng, kích thước hình ảnh, diện mạo, biểu hiện cảm xúc của khuôn mặt hay ảnh hưởng của tham số môi trường

Để xây dựng một hệ thống nhận dạng khuôn mặt có đầu vào của hệ thống là một hình ảnh kỹ thuật số hay một khung hình video từ một nguồn video. Đầu ra là xác định hoặc xác minh người ở trong bức hình hoặc trong video đó là ai. Hướng tới mục tiêu này chúng ta thường chia thủ tục nhận dạng khuôn mặt gồm ba bước: Phát hiện khuôn mặt, trích rút đặc trưng và nhận dạng khuôn mặt.



Hình 1.1 Hệ thống nhận dạng khuôn mặt

Phát hiện khuôn mặt (Face Detection): Chức năng chính của bước này là phát hiện ra khuôn mặt xem nó có xuất hiện ở trong một bức hình hay một đoạn video hay không? Tỉ lệ phát hiện ra khuôn mặt phụ thuộc nhiều vào điều kiện về độ sáng, hướng khuôn mặt, biểu hiện cảm xúc trên khuôn mặt hay các yếu tố môi trường khác. Để hệ thống nhận dạng hoạt động đạt hiệu quả cao thì hình ảnh khuôn mặt sau khi được phát hiện cần chuẩn hóa về kích thước, ánh sáng.

Trích rút đặc trưng (Feature Extraction): Sau khi phát hiện ra khuôn mặt trong bức ảnh, chúng ta tiến hành trích rút những đặc trưng của khuôn mặt. Bước này trích xuất ra một vector đặc trưng đại diện cho một khuôn mặt. Nó phải đảm bảo được tính duy nhất của một khuôn mặt

Nhận dạng khuôn mặt (Face Recognition): Với hình ảnh đầu vào sau khi phát hiện ra khuôn mặt, trích rút các đặc trưng của khuôn mặt và đem so sánh các đặc trưng này với cơ sở dữ liệu khuôn mặt.

Bài toán nhận dạng khuôn mặt được ứng dụng nhiều trong các lĩnh vực đời sống đặc biệt ở những lĩnh vực công nghệ cao, yêu cầu về an ninh, bảo mật. Do đó để hệ thống nhận dạng khuôn mặt hoạt động mạnh mẽ với tốc độ và độ tin cậy thì có rất nhiều các phương pháp về nhận dạng khuôn mặt được đưa ra. Các phương pháp có thể được phân loại theo các tiêu chí khác nhau như nhận dạng với dữ liệu ảnh đầu vào là ảnh tĩnh 2D (Elastic Bunch Graph, Active Appearance Model). Phương pháp này là phổ biến nhất và tương lai sẽ là 3D (3D Morphable Model). Tuy nhiên trên thực tế người ta hay chia phương pháp nhận dạng khuôn mặt ra thành 2 loại:

- Nhận dạng dựa trên các đặc trưng của các phần tử trên khuôn mặt (Feature Base Face Recognition)

- Nhận dạng dựa trên xét tổng thể toàn khuôn mặt (Appearance Based Face Recognition).

## Tổng quan về hệ thống

Mô tả: Hệ thống camera tự động chụp hình ảnh, chương trình sẽ dùng thuật toán nhận diện khuôn mặt của thư viện OpenCV để nhân diện những khuôn mặt đã được chụp bởi camera

- Trình tự thực hiện:

1. Camera chụp hình ảnh.

2. Chương trình tiến hành phát hiện khuôn mặt và lưu hình ảnh khuôn mặt

3. Chương trình tiến hành phân tích dữ liệu và đưa vào mô hình đào tạo

4. Lưu cơ sở dữ liệu thông tin đã nhận diện.

- Kết quả: Nhận diện khuôn mặt và điểm danh sự có mặt

## Các công cụ

Để giải quyết bài toán nhận dạng khuôn mặt trong đồ án này em sử dụng thư viện opencv và ngôn ngữ Python trên phần mềm Visual Studio Code.

### Python

Python là ngôn ngữ lập trình hướng đối tượng, cấp cao, mạnh mẽ, được tạo ra bởi Guido van Rossum. Nó dễ dàng để tìm hiểu và đang nổi lên như một trong những ngôn ngữ lập trình nhập môn tốt nhất cho người lần đầu tiếp xúc với ngôn ngữ lập trình. Python hoàn toàn tạo kiểu động và sử dụng cơ chế cấp phát bộ nhớ tự động. Python có cấu trúc dữ liệu cấp cao mạnh mẽ và cách tiếp cận đơn giản nhưng hiệu quả đối với lập trình hướng đối tượng. Cú pháp lệnh của Python là điểm cộng vô cùng lớn vì sự rõ ràng, dễ hiểu và cách gõ linh động làm cho nó nhanh chóng trở thành một ngôn ngữ lý tưởng để viết script và phát triển ứng dụng trong nhiều lĩnh vực, ở hầu hết các nền tảng.



Hình 1.2 Ngôn ngữ Python

Python là một ngôn ngữ lập trình thông dịch (interpreted), hướng đối tượng (object - oriented), và là một ngôn ngữ bậc cao (high - level) ngữ nghĩa động (dynamic semantics). Python hỗ trợ các module và gói (packages), khuyến khích chương trình module hóa và tái sử dụng mã. Trình thông dịch Python và thư viện chuẩn mở rộng có sản dưới dạng mã nguồn hoặc dạng nhị phân miễn phí cho tất cả các nền tảng chính và có thể được phân phối tự do.

Các đặc điểm của Python:

* Ngữ pháp đơn giản, dễ đọc.
* Vừa hướng thủ tục (Procedural - oriented), vừa hướng đối tượng (object - oriented)
* Hỗ trợ module và hỗ trợ gói (package)
* Xử lý lỗi bằng ngoại lệ (Exception)
* Kiểu dữ liệu động ở mức cao.
* Có các bộ thư viện chuẩn và các module ngoài, đáp ứng tất cả các nhu cầu lập trình.
* Có khả năng tương tác với các module khác viết trên C / C ++ (Java cho Jython, hoặc .Net cho IronPython). Có thể nhúng vào ứng dụng như một giao tiếp kịch bản (scripting interface).

### Thư viện OpenCV

Opencv (Open Computer Vision library) do Intel phát triển, được giới thiệu năm 1999 và hoàn thiện thành phiên bản 1.0 năm 2006. Thư viện opencv - gôm khoảng 500 hàm - được viết bằng ngôn ngữ lập trình C và tương thích với các hệ điều hành Windows, Linux, Mac OS ... đóng vai trò xác lập chuẩn giao tiếp, dữ liệu, thuật toán cho lĩnh vực CV và tọa điều kiện cho mọi người tham gia nghiên cứu và phát triển ứng dụng.

Trước Opencv không có một công cụ chuẩn nào cho lĩnh vực xử lý ảnh. Các đoạn code đơn lẻ do các nhà nghiên cứu tự viết thường không thống nhất và không ổn định. Các bộ công cụ thương mại như Matlab, Simulink, … lại có giá cao chỉ thích hợp cho các công ty phát triển các ứng dụng lớn. Ngoài ra còn có các giải pháp kèm theo thiết bị phần cứng mà phần lớn là mã đóng và được thiết kế riêng cho từng thiết bị, rất khó khăn cho việc mở rộng ứng dụng.

OpenCV là công cụ hữu ích cho những người bước đầu làm quen với xử lí ảnh số vì các ưu điểm sau:

* OpenCV là công cụ chuyên dụng: được Intel phát triển theo hướng tối ưu hóa cho các ứng dụng xử lí và phân tích ảnh, với cấu trúc dữ liệu hợp lí, thư viện tạo giao diện, truy xuất thiết bị phần cứng được tích hợp sẵn. OpenCV thích hợp để phát triển nhanh ứng dụng
* OpenCV là công cụ mã nguồn mở: Không chỉ là công cụ miễn phí, việc được xây dựng trên mã nguồn mở giúp OpenCV trở thành công cụ thích hợp cho nghiên cứu và phát triển, với khả năng thay đổi và mở rộng các mô hình, thuật toán
* OpenCV đã được sử dụng rộng rãi: Từ năm 1999 đến nay, OpenCV đã thu hút được một lượng lớn người dùng, trong đó có các công ty lớn như Microsoft, IBM, Sony, Siemens, Google và các nhóm nghiên cứu ở Stanford, MIT, CMD, Cambridge, ... Nhiều forum hỗ trợ và Cộng đồng người dùng đã được thành lập, tạo nên kênh thông tin rộng lớn, hữu ích cho việc tham khảo, tra cứu

### Visual Studio Code

Visual Studio Code là một trình biên tập mã dược phát triển bởi Microsoft dành cho Windows, Linux và macOS. Nó hỗ trợ chức năng debug, đi kèm với Git, có syntax highlighting, tự hoàn thành mã thông minh, snippets, và cải tiến mã nguồn. Visual Studio Code là một trình biên tập mã. Nó hỗ trợ nhiều ngôn ngữ và chức năng tùy vào ngôn ngữ sử dụng theo như trong bảng sau. Nhiều chức năng của Visual Studio Code không hiển thị ra trong các menu tùy chọn hay giao diện người dùng. Thay vào đó, chúng được gọi thông qua khung nhập lệnh hoặc qua một tập tin .json ( ví dụ như tập tin tùy chỉnh của người dùng ). Khung nhập lệnh là một giao diện theo dòng lệnh. Tuy nhiên, nó biến mất khi người dùng nhấp bất cứ nơi nào khác, hoặc nhấn tổ hợp phím để tương tác với một cái gì đó ở bên ngoài đó. Tương tự như vậy với những dòng lệnh tổn nhiều thời gian để xử lý. Khi thực hiện những điều trên thì quá trình xử lý dòng lệnh đó sẽ bị hủy

### SQLite

SQLite là hệ thống cơ sở dữ liệu quan hệ nhỏ gọn, hoàn chỉnh, có thể cài đặt bên trong các trình ứng dụng khác. SQLite được Richard Hipp viết dưới dạng thư viện bằng ngôn ngữ lập trình C. SQLite có các ưu điểm sau:

* Tin cậy: các hoạt động transaction (chuyển giao) nội trong cơ sở dữ liệu được thực hiện trọn vẹn, không gây lỗi khi xảy ra sự cố phần cứng
* Tuân theo chuẩn SQL92 (chỉ có một vài đặc điểm không hỗ trợ)
* Không cần cài đặt cấu hình
* Kích thước chương trình gọn nhẹ, với cấu hình đầy đủ chỉ không đầy 300 kB
* Thực hiện các thao tác đơn giản nhanh hơn các hệ thống cơ sở dữ liệu khách/chủ khác
* Không cần phần mềm phụ trợ
* Phần mềm tự do với mã nguồn mở, được chú thích rõ ràng

## Cấu trúc báo cáo

Chương 1: giới thiệu tổng quan

Nội dung chương này giới thiệu bài toán nhận dạng khuôn mặt, cách triển khai giải quyết bài toán và các công cụ

Chương 2: Thuật toán nhận dạng

Nội dung chương này diễn giải những hiểu biết cá nhân về thuật toán nhận diện khuôn mặt được sử dụng trong chia trình thông qua các hàm của thư viện

Chương 3: Phân tích thiết kế và triển khai

Nội dung chưa này thuật lại quá trình viết code cũng như các kết quá đạt được trong từng bước thông qua code và hình ảnh minh họa

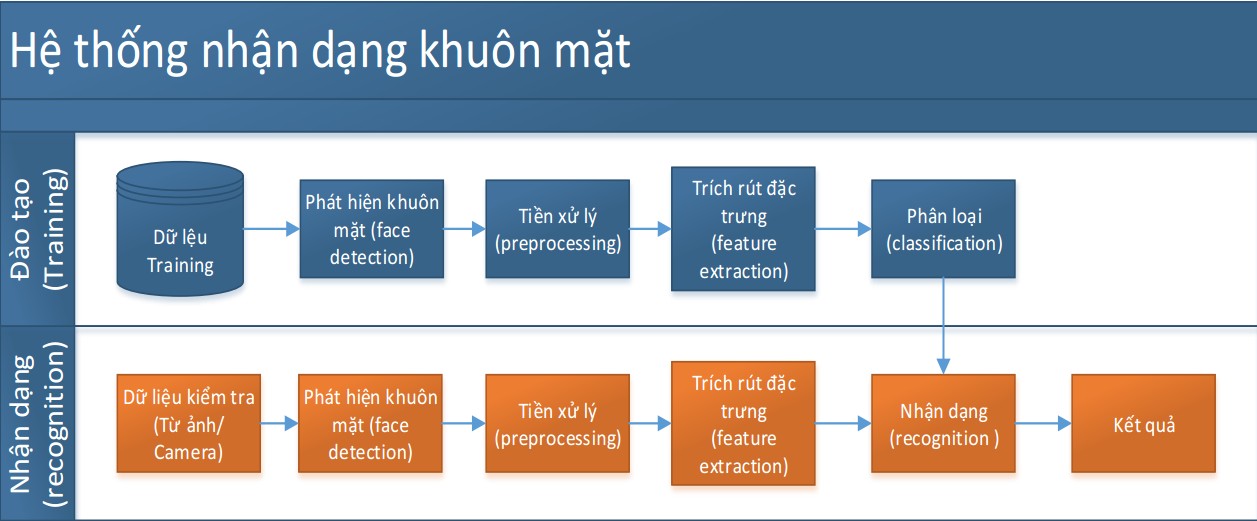
Chương 4: Kết luận và hướng phát triển

Nội dung chương này sẽ kết luận, đánh giá quá trình nghiên cứu và hướng phát triển cho chương trình sau này.

# KỸ THUẬT NHẬN DẠNG KHUÔN MẶT

## Phát hiện khuôn mặt

Phát hiện khuôn mặt (Face Detection) là một kĩ thuật máy tính để xác định được các vị trí và các kích thước của khuôn mặt người trong các ảnh bất kỳ. Kỹ thuật này nhận biết các đặc trưng của khuôn mặt và bỏ qua những thứ khác như: tòa nhà, cây cối, cơ thể.



Hình 2.1 Hệ thống nhận diện khuôn mặt

### Vấn đề

Như ở não của con người chúng ta có những sợi thần kinh để làm tất cả các việc đó một cách tự động và ngay lập tức. Trong thực tế, con người rất giỏi trong việc nhận diện khuôn mặt và tưởng tượng ra các khuôn mặt trong các vật thể hằng ngày.

Máy tính thì lại không có khả năng bậc cao kiểu đó ít nhất là hiện tại (trong tương lai có thể có). Nên chúng ta cần dạy cho chúng cách để làm từng bước riêng biệt trong quá trình nhận dạng. Chúng ta cần xây dựng một quy trình (hệ thống) nơi chúng ta giải quyết từng bước của nhận diện khuôn mặt một cách riêng biệt, và chuyển kết quả hiện tại đó cho bước tiếp theo.

### Tìm khuôn mặt

Mục tiêu bước này là chúng ta sẽ tìm vị trí của khuôn mặt trong tấm hình trước khi chúng ta phân biệc chúng và qua bước tiếp theo.

Nếu bạn đang sử dụng bất kì camera nào trong khoảng 5 năm trở lại đây thì bạn có thể thấy nhận diện khuôn mặt được tích hợp sẵn trong thiết bị. Nhận diện khuôn mặt là một tính năng tuyệt vời cho máy ảnh. Khi máy ảnh có thể tự động chọn ra các khuôn mặt, chúng có thể chắc chắn rằng tất cả các khuôn mặt được là nét trước khi nó ghi hình. Nhưng chúng ta lại sử dụng nó với mục đích khác, tìm kiếm vị trí khuôn mặt trong tấm hình và gửi kết quả đó cho bước tiếp theo trong quá trình

Nhận diện khuôn mặt đã trở thành xu hướng vào đầu những năm 2010 khi Paul Viola và Michael Jones phát minh ra cách để nhận diện khuôn mặt với tốc độ đủ để chạy trên con các dòng máy ảnh rẻ tiền. Tuy nhiên thì còn có các phương pháp đáng tin cậy cũng đã xuất hiện. Chúng ta sử dụng phương pháp được phát minh năm 2005 được gọi là “Histogram of Oriented Gradients” (rút gọn thành HOG)

Để tìm những khuôn mặt trong một tấm hình, chúng ta bắt đầu với việc biến tấm ảnh của chúng ta thành tấm ảnh cảm, chỉ có đen và trắng vì chúng ta không cần màu sắc để tìm khuôn mặt. Sau đó chúng ta nhìn vào từng pixel trong tấm hình cùng một lúc. Với mỗi pixel một, chúng ta lại nhìn vào những pixel lân cận nó.

Mục đích của chúng ta là tìm ra pixel hiện tại có màu tối như thế nào so với các pixel lân cận nó. Khi đó chúng ta sẽ về một mũi tên theo chiều mà màu trở nên tới hơn.

Nếu bạn lặp lại tiến trình đó với mỗi pixel một trong trong tấm ảnh, bạn sẽ kết thúc với mỗi pixel được thay thế bởi một mũi tên. Những mũi tên đó được gọi là “gradients” (vectơ độ dốc) và chúng chỉ ra dòng chảy lưu lượng) từ những pixel sáng đến những Pixel tối trên toàn bộ hình ảnh. Điều này có vẻ là một việc làm ngẫu nhiên, nhưng đó là một lý do tốt để thay thế các pixel đó thành gradients. Nếu chúng ta phân tích trực tiếp các pixel tối và Các pixel sáng trong bức ảnh của cùng một người sẽ nhận được tổng các giá trị hoàn toàn khác nhau.

Nhưng nếu xem xét hướng sáng thay đổi, cả hai hình ảnh tối và hình ảnh sáng sẽ cho kết quả với cùng một đại diện (con số) chính xác. Điều đó làm cho vấn đề để giải quyết hơn. Những việc lưu gradient cho mỗi pixel đơn lẻ cho chúng ta quá nhiều chi tiết Sẽ tốt hơn nếu chúng ta có thể nhìn thấy dòng chảy sáng tối cơ bản ở mức cao hơn để chúng ta có thể thấy mô hình cơ bản của hình ảnh.

Để làm điều này, chúng tôi sẽ chia hình ảnh thành các ô vuông nhỏ 16x16 Pixel mỗi hình. Trong mỗi ô vuông, chúng tôi se để xem có bao nhiêu độ dốc theo từng hướng chính (có bao nhiêu điểm hướng lên, hướng lên phải, điểm phải vv ...). Sau đó, chúng tôi sẽ thay thế hình vuông đó trong hình ảnh bằng các hướng mũi tên nổi bật nhất

Kết quả cuối cùng là chúng ta biến hình ảnh gốc thành một hình đại diện rất đơn giản, nắm bắt cấu trúc cơ bản của khuôn mặt một cách đơn giản hơn.

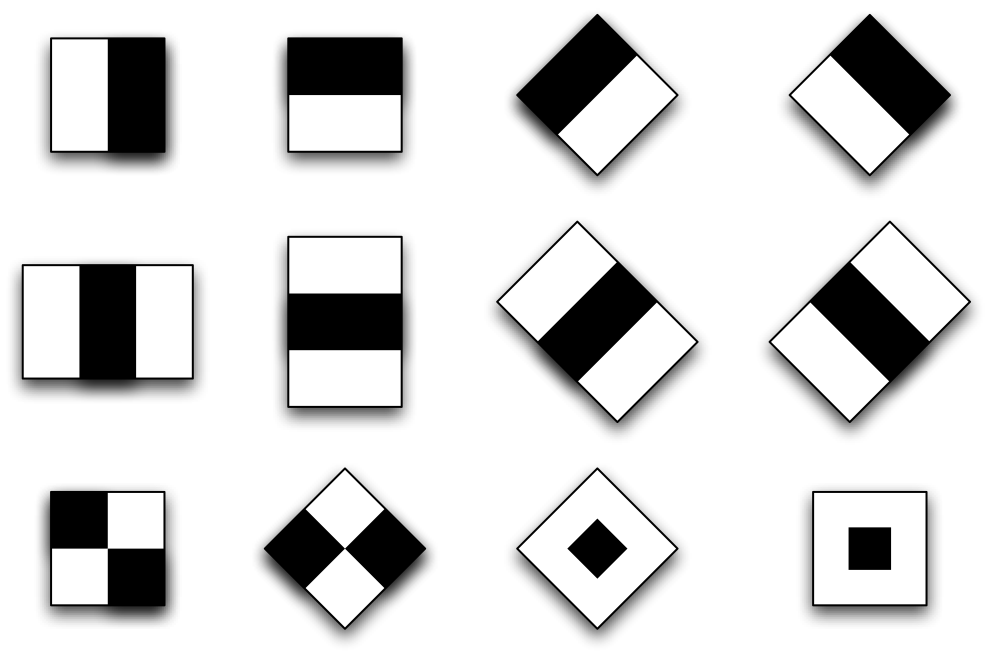
Để tìm các khuôn mặt trong hình ảnh HOG này, tất cả những gì chúng ta phải làm là tìm phần hình ảnh của chúng ta trông giống nhất với mẫu HOG đã biết được trích xuất từ một loạt các khuôn mặt đào tạo khác.

### Phương pháp nhận diện khuôn mặt của Violas & John

Phát hiện mặt người là bài toán cơ bản được xây dựng từ nhiều năm nay, có nhiều phương pháp được đưa ra như sử dụng template matching, neuron network…Cho tới nay bài toán này hầu như được giải quyết dựa trên phương pháp sử dụng các đặc trưng haar like. Phương pháp này được cho là đơn giản và kết quả phát hiện là tương đối cao, lên tới 98%, các hãng sản xuất máy ảnh như Canon, Samsung… cũng đã tích hợp nó vào trong các sản phẩm của mình. Và trong bài viết này, mình sẽ cùng các bạn tìm hiểu về phương pháp pháp phát hiện khuôn mặt của Violas và John được cài đặt trong OpenCV

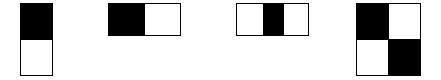
**Các đặc trưng Haar-Like**

Các đặc trưng Haar-Like là những hình chữ nhật được phân thành các vùng khác nhau như hình



Hình 2.2 Đặc trưng Haar-Like

Đặc trưng do Viola và Jones công bố gồm 4 đặc trưng cơ bản để xác định khuôn mặt người. Mỗi đặc trưng Haar-Like là sự kết hợp của hai hay ba hình chữ nhật trắng hay đen như trong hình sau



Hình 2.3 Bốn đặc trưng cơ bản

Để sử dụng các đặc trưng này vào việc xác định khuôn mặt người, 4 đặc trưng Haar-Like cơ bản được mở rộng ra và được chia làm 3 tập đặc trưng như sau:

-Đặc trưng cạnh (edge feature) 

-Đặc trưng đường(line feature) 

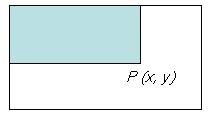
-Đặc trưng xung quanh tâm(center-surround features) 

Dùng các đặc trưng trên, ta có thể tính được các giá trị của đặc trưng Haar-Like là sự chênh lệch giữa tổng của các pixel của vùng đen và vùng trắng như trong công thức sau:



Hình 2.4 Công thức tính trong Haar-Like

Viola và Joines đưa ra một khái niệm gọi là Integral Image, là một mảng 2 chiều với kích thước bằng với kích thước của ảnh cần tính đặc trưng Haar-Like, với mỗi phần tử của mảng này được tính bằng cách tính tổng của điểm ảnh phía trên (dòng-1) và bên trái (cột-1) của nó.



Hình 2.5 Integral Image

Công thức tính Intergral Image:

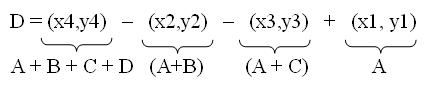


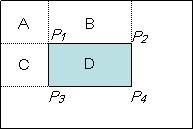
Sau khi tính được Integral Image, việc tính tổng các giá trị mức xám của một vùng bất kỳ nào đó trên ảnh thực hiện rất đơn giản theo cách sau:

Giả sử ta cần tính tổng giá trị mức xám của vùng D như hình dưới, ta có thể tính được như sau:

D = A + B + C + D – (A+B) – (A+C) + A

Với A + B + C + D chính là giá trị tại điểm P4 trên Integral Image, tương tự như vậy A+B là giá trị tại điểm P2, A+C là giá trị tại điểm P3, và A là giá trị tại điểm P1. Vậy ta có thể viết lại biểu thức tính D ở trên như sau:



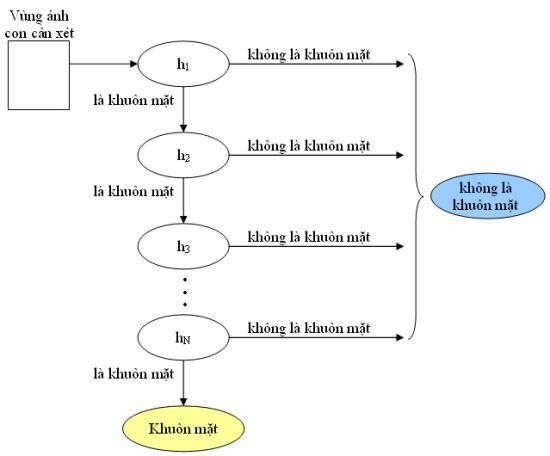


Hình 2.6 Tính tổng mức xám vùng D

**AdaBoost**

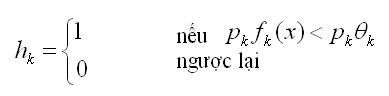
AdaBoost là một bộ phân loại mạnh phi tuyến phức dựa trên hướng tiếp cận boosting được Freund và Schapire đưa ra vào năm 1995. Adaboost cũng hoạt động trên nguyên tắc kết hợp tuyến tính các weak classifiers để hình thành một trong các classifiers.

Viola và Jones dùng AdaBoost kết hợp các bộ phân loại yếu sử dụng các đặc trưng Haar-like theo mô hình phân tầng (cascade) như sau:

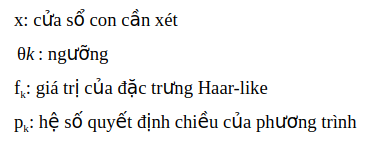


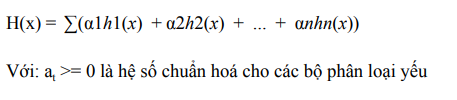
Hình 2.7 AdaBoost

Trong đó, h(k) là các bộ phân loại yếu, được biểu diễn như sau:

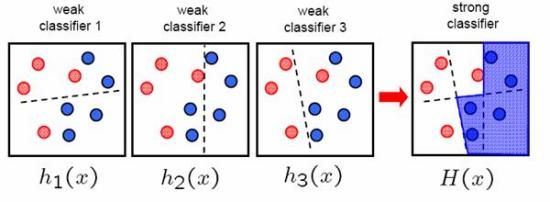


với:



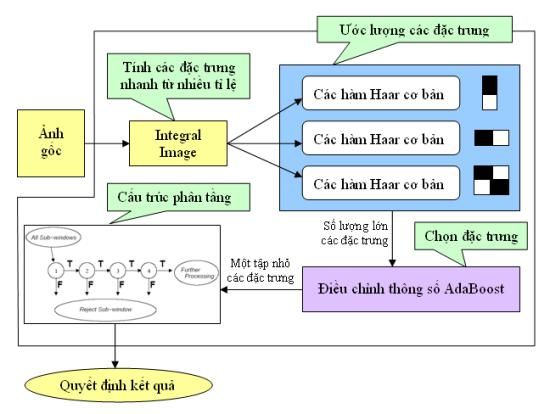
AdaBoost sẽ kết hợp các bộ phân loại yếu thành bộ phân loại mạnh như sau: 

Đây là hình ảnh minh họa việc kết hợp các bộ phân loại yếu thành bộ phân loại mạnh



Hình 2.8 Kết hợp bộ phân loại yếu thành mạnh

**Sơ đồ phát hiện khuôn mặt:**  Sau đây là sơ đồ thể hiện việc áp dụng vào để phát hiện khuôn mặt



Hình 2.9 Sơ đồ phát hiện khuôn mặt

## Tiền xử lý ảnh

Bước này nhằm mục đích lọc nhiễu, nâng cao chất lượng ảnh, trong bước này bao gồm các bước: Căn chỉnh ảnh, chuẩn hóa ánh sáng

## Trích rút đặc trưng

Trích rút đặc trưng là kỹ thuật sử dụng các thuật toán để lấy ra những thông tin mang những đặc điểm riêng biệt của một người. Principle Components Analysis - PCA là một thuật toán được sử dụng để tạo ra một hình ảnh mới từ hình ban đầu. Ảnh mới này có kích thước nhỏ hơn rất nhiều so với ảnh ban đầu và vẫn mang những đặc trưng cơ bản nhất của ảnh cần nhận dạng.

Về bản chất, PCA tìm ra một không gian mới theo hướng biến thiên mạnh nhất của một tập hợp các vector trong không gian cho trước. Trong không gian mới này người ta hy vọng rằng việc phân loại sẽ mang lại kết quả tốt hơn so với không gian ban đầu.

Không gian mới được tạo bởi được tạo bởi PCA được cấu thành từ K vector đơn vị có chiều là N. Mỗi vector được gọi là Eigenfaces

A= =>B== với N<<K

Theo công thức: W = T.A

Với T là ma trận chuyển đổi, T có kích thước K x N

## Dữ liệu hệ thống nhận dạng

Dữ liệu hệ thống nhận dạng được chia làm 3 tập

* Tập huấn luyện (training set),
* Tập tham chiếu (reference set)
* Tập để nhận dạng (probe set).

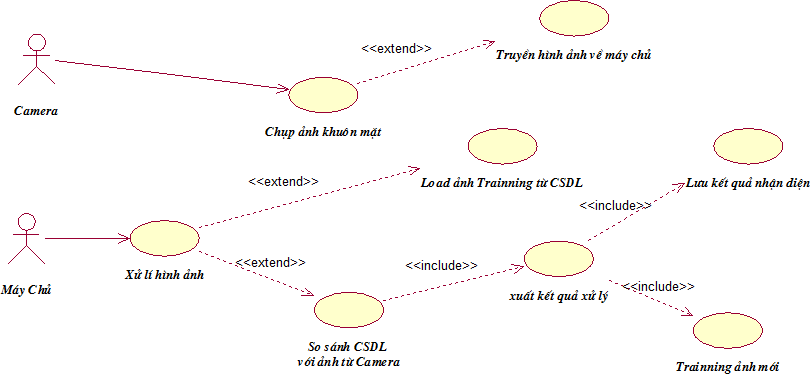
Trong nhiều hệ thống, tập training trùng với tập reference (trong báo cáo này tập huấn luyện trùng với tập training). Tập training gồm các ảnh được dùng để huấn luyện, thông thường tập này được dùng để sinh ra một không gian con là một ma trận và phương pháp hay được sử dụng là PCA, LDA…Tập reference gồm các ảnh đã biết danh tính được chiếu vào không gian con ở bước training ... Sau khi thực hiện chiếu tập reference vào không gian con, hệ thống lưu lại kết quả là một ma trận với mỗi cột của ma trận là một vector tương ứng với ảnh (định danh đã biết) để thực hiện nhận dạng (hay phân lớp).

# PHÂN TÍCH THIẾT KẾ VÀ TRIỂN KHAI THỰC HIỆN

## Phân tích thiết kế hệ thống

### Sơ đồ Use-case

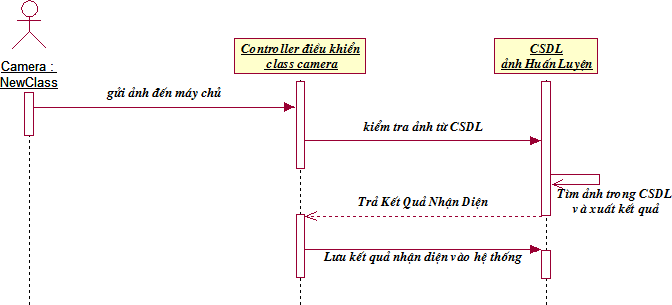
Sơ đồ use case mô tả chức năng của 2 actor Camera và Máy chủ.



Hình 3.1 Use case Camera và Chương trình

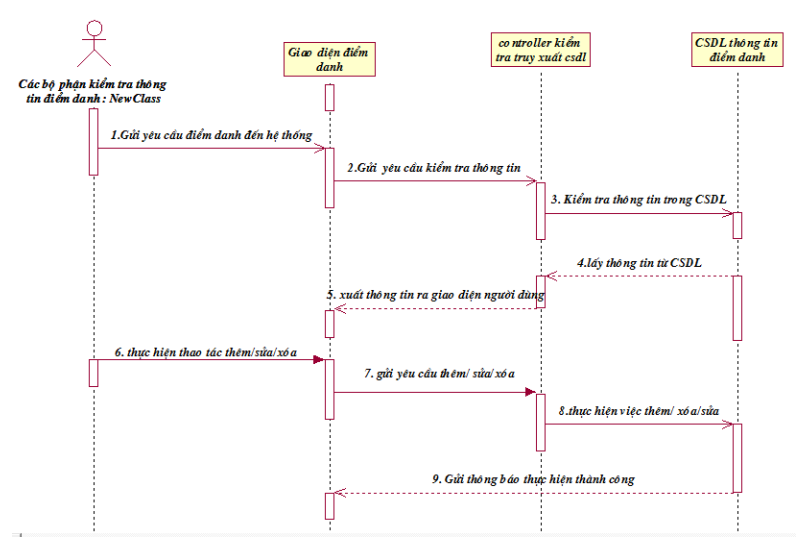
### Mô hình sequence diagram

Mô hình chức năng nhận diện camera



Hình 3.2 Mô hình chức năng nhận diện camera

Dòng sự kiện: Camera nhận diện hình ảnh gửi về Controller xử lý, kiểm tra xem hình ảnh người cần nhận diện đã có trong CSDL chưa, nếu có rồi thì trả kết quả nhận diện, nếu không có thì controller nhận diện và lưu hệ thống CSDL



Hình 3.3 Chức năng thông báo, chỉnh sửa danh sách điểm danh

Dòng sự kiện: Các bộ phận thực hiện thao tác kiểm tra thông tin điểm danh trên giao diện điểm danh sinh viên, sau đó controller sẽ thực hiện yêu cầu kiểm tra xem thông tin cần lấy trong CSDL nếu có thì xuất thông tin về giao diện. Các bộ phận tiếp tục từ thông tin đã nhận được thực hiện các thao tác thêm xóa sửa, controller điều khiển nhận các yêu cầu và thực hiện thao tác trên CSDL. Sau đó thông báo kết qua ra giao diện.

## Cài đặt môi trường

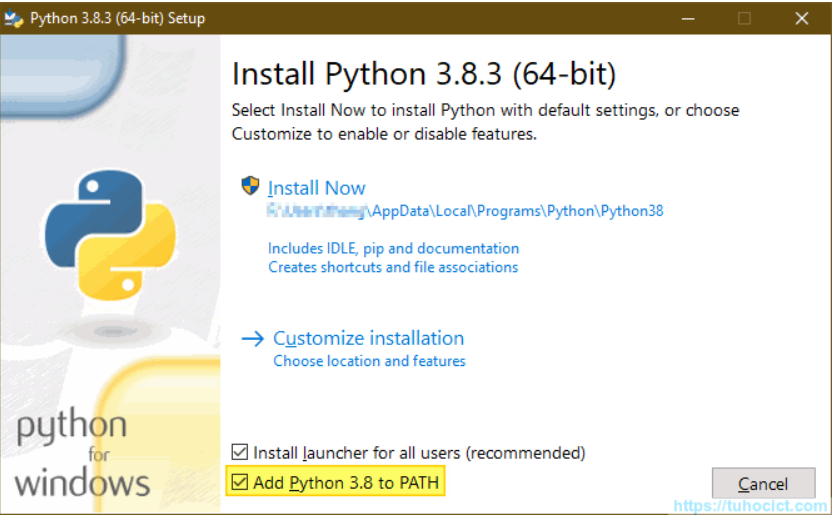
### Python

Để có thể làm việc với Python (viết / thực thi ứng dụng) bạn cần cài đặt Python. Do Python hoạt động đa nền tảng, quá trình cài đặt phụ thuộc vào hệ điều hành.

Để đơn giản hóa việc minh họa, chúng ta sẽ chỉ sử dụng Windows. Việc cài đặt Python trên các hệ điều hành khác sẽ trình bày trong một số bài blog riêng.

Tải bản cài đặt Python cho Windows từ link https://www.python.org/downloads/Windows/ .

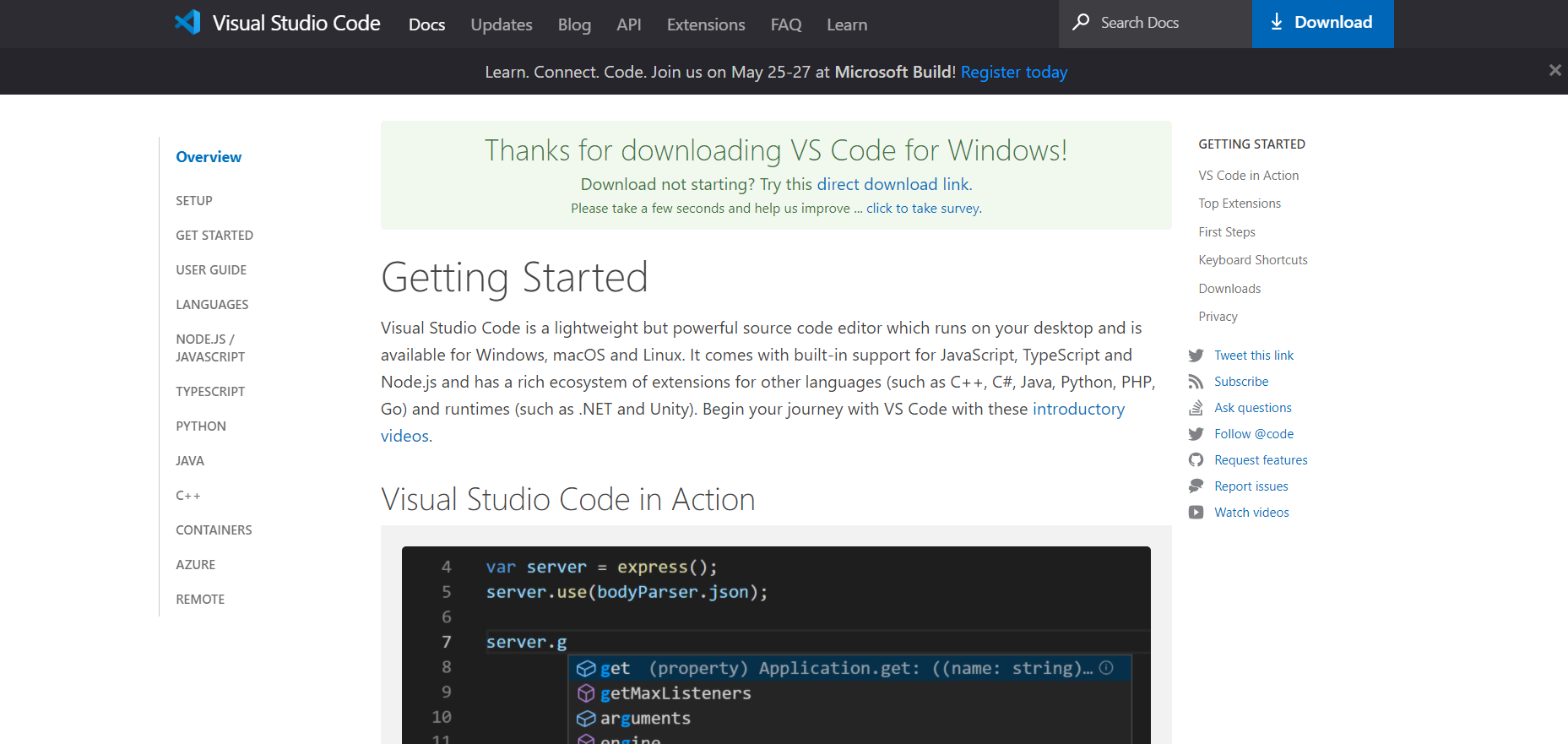
Bạn có thể lựa chọn bàn cài đặt offline (executable installer) hoặc cài từ web (web - based installer) 32 hoặc 64 bit. Lưu ý Python không hỗ trợ windows XP. Khi chạy chương trình cài đặt lưu ý chọn Add Python to PATH để dễ dàng chạy Python từ dấu nhắc lệnh.



Hình 3.4 Cài đặt Python

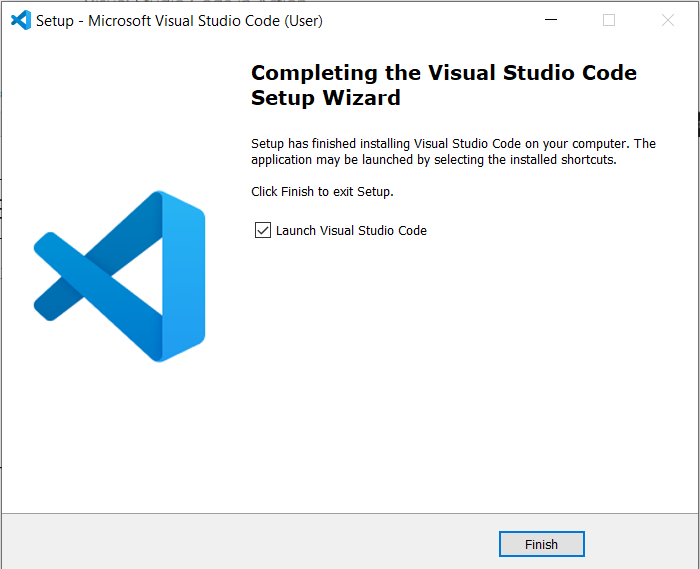
### Visual studio code

Đầu tiên tải Visual Studio Code ở trang chủ.



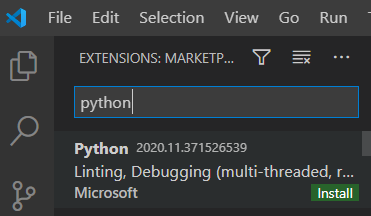
Hình 3.5 Tải Visual Studio Code

Giao diện sau khi các bạn cài xong:



Hình 3.6 Cài đặt Visual Studio Code

Bây giờ bắt đầu cài extension và sử dụng nó cho python. Bấm: Ctrl + Shift + x. Tìm python bấm install.



Hình 3.7 Thêm Python vào phần mềm

Sau đó bấm ctrl + n để tạo mới 1 trang code sau đó bấm: Ctrl + Shift + p để chọn môi trường lập trình.

### Cài đặt OpenCV

Ở đây để đơn giản cho việc cài đặt thì chúng ta sử dụng một thư viện có sẵn trên kho quản lý gói của python là opencv-python. Đây là một thư viện không phải của tổ chức OpenCV tuy nhiên nó có tính năng y hệt và được cập nhật thường xuyên theo sự thay đổi của OpenCV. Hay nói cách khác đây là một đóng gói cho OpenCV trên kho thư viện của python. Nếu bạn muốn sử dụng opencv phiên bản gốc thì lên trang sourceforge.net chọn phiên bản rồi tải xuống. Tiếp theo là giải nén và cài đặt bằng tay theo hướng dẫn. Vì cách cài đặt này để cái đặt cho nhiều ngôn ngữ khác nhau nên cách này không khuyến khích cho người vừa bắt đầu sử dụng opencv.

Sử dụng đoạn lênh sau để cài đặt gói từ kho thư viện pip:

*pip install opencv-python*

Để kiểm tra cài đặt thành công hay không ta thực hiện đoạn lênh sau để kiểm tra phiên bản của opencv-python. Tùy từng thời điểm thì phiên bản của gói sẽ khác nhau. Các phiên bản mới là cập nhật cho opencv mới hơn.

*import cv2*

*print(cv2.\_\_version\_\_)*

## Xây dựng phần mềm

**3.3.1** **Giao diện ứng dụng**

**a) Giao đăng nhập của ứng dụng**

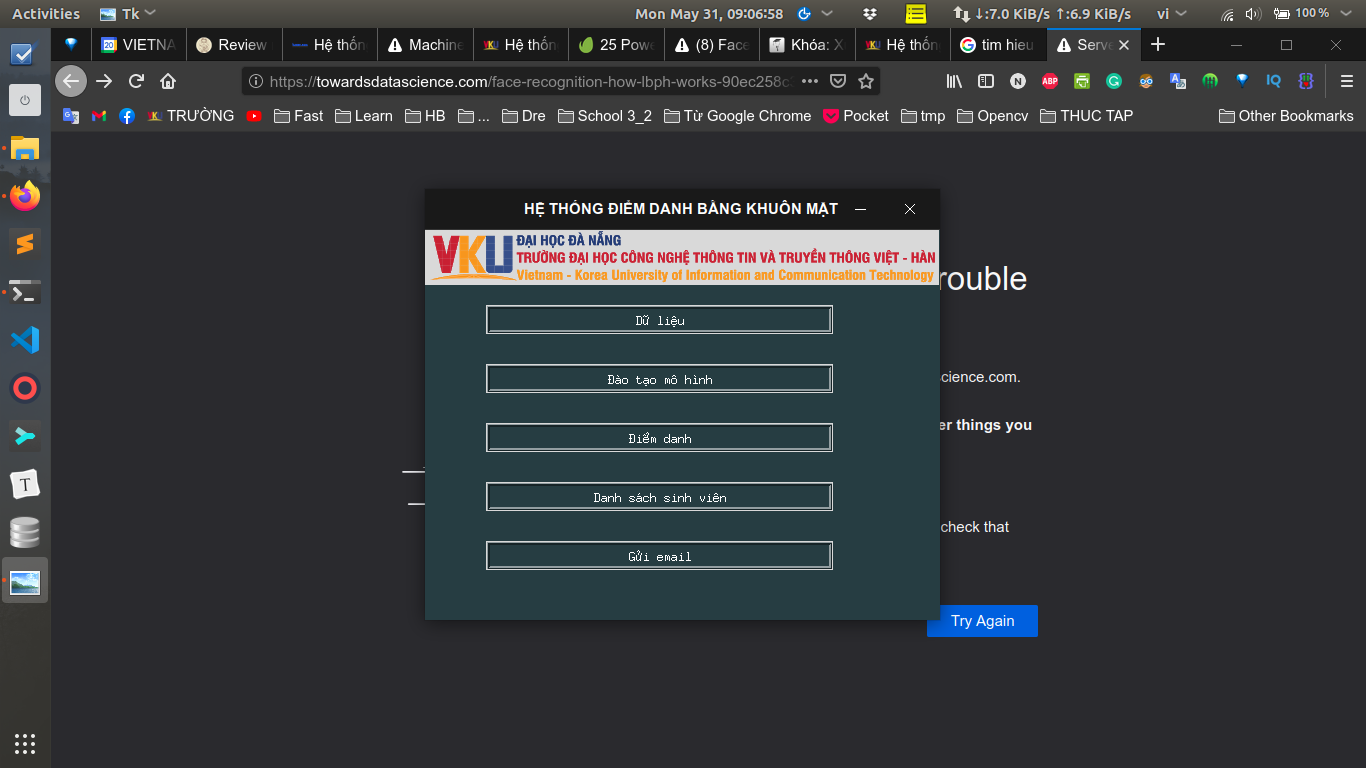
Để đảm bảo tính bảo mật hệ thống, chúng em thiết kế cơ sỡ dữ liệu để đăng nhập hệ thống điểm danh. Chỉ những người được cấp tài khoản mới có thể vào được chương trình để xem và thực hiện các chức năng của phần mềm



Hình 3.8 Giao diện đăng nhập

**b) Giao diện bảng điều khiển chương trình**

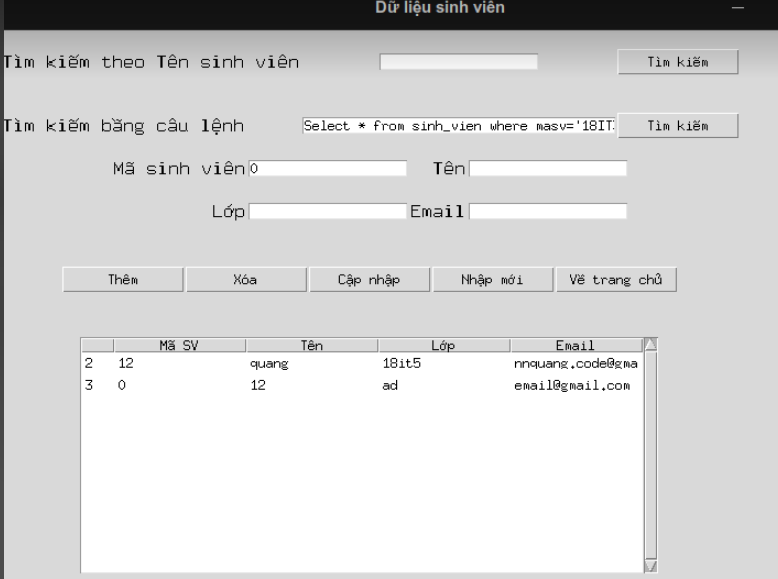
Bảng điều khiển chứa các chức năng chính của chương trình, phục vụ cho mục đích xem tổng thể các chức năng chính của phần mềm



Hình 3.9 Bảng điều khiển

**c) Giao diện quản lý thông tin sinh viên**

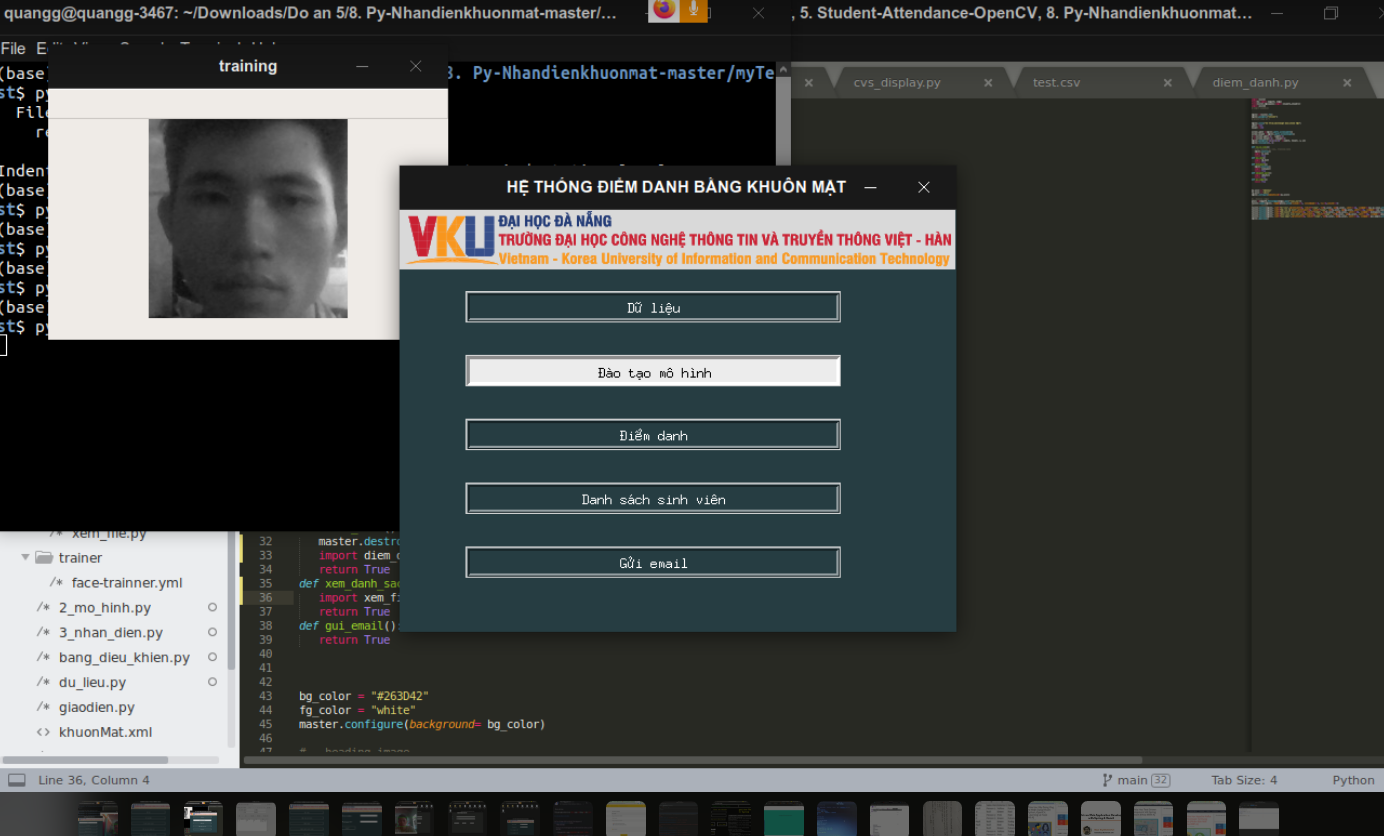
Đây là chức năng thao tác với cơ sở dữ liệu của sinh viên, gồm có các chức năng như thêm, sữa, xoá, cập nhật và tìm kiếm thông tin. Sau khi đã điền đầy đủ thông tin để thêm sinh viên thì hệ thống sẽ tiến hành chụp 100 tấm hình của sinh viên đó, tên file sẽ chứa mã sinh viên cùng thứ tự của bức hình, dữ liệu sẽ được lưu trữ trong thư mục hình ảnh sinh viên



Hình 3.10 Quản lý thông tin sinh viên

**d) Giao diện đào tạo mô hình để nhận diện**

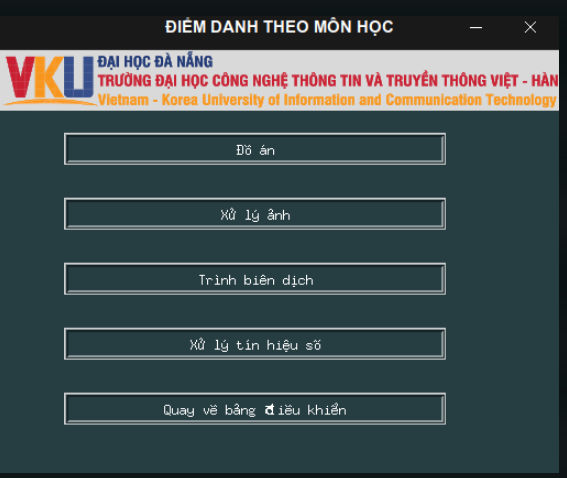
Sau khi đã lấy được các hình ảnh của sinh viên thì hệ thống sẽ tiến hành đào tạo mô hình nhận điện. Dữ liệu đầu vào của mô hình bao gồm 100 tấm hình của sinh viên và mã sinh viên của những tấm hình đó. Sau khi quá trình đào tạo mô hình hoàn tất thì sẽ lưu dữ liệu vào một file yml để phục vụ nhanh chóng quá trình nhận dạng khuôn mặt người



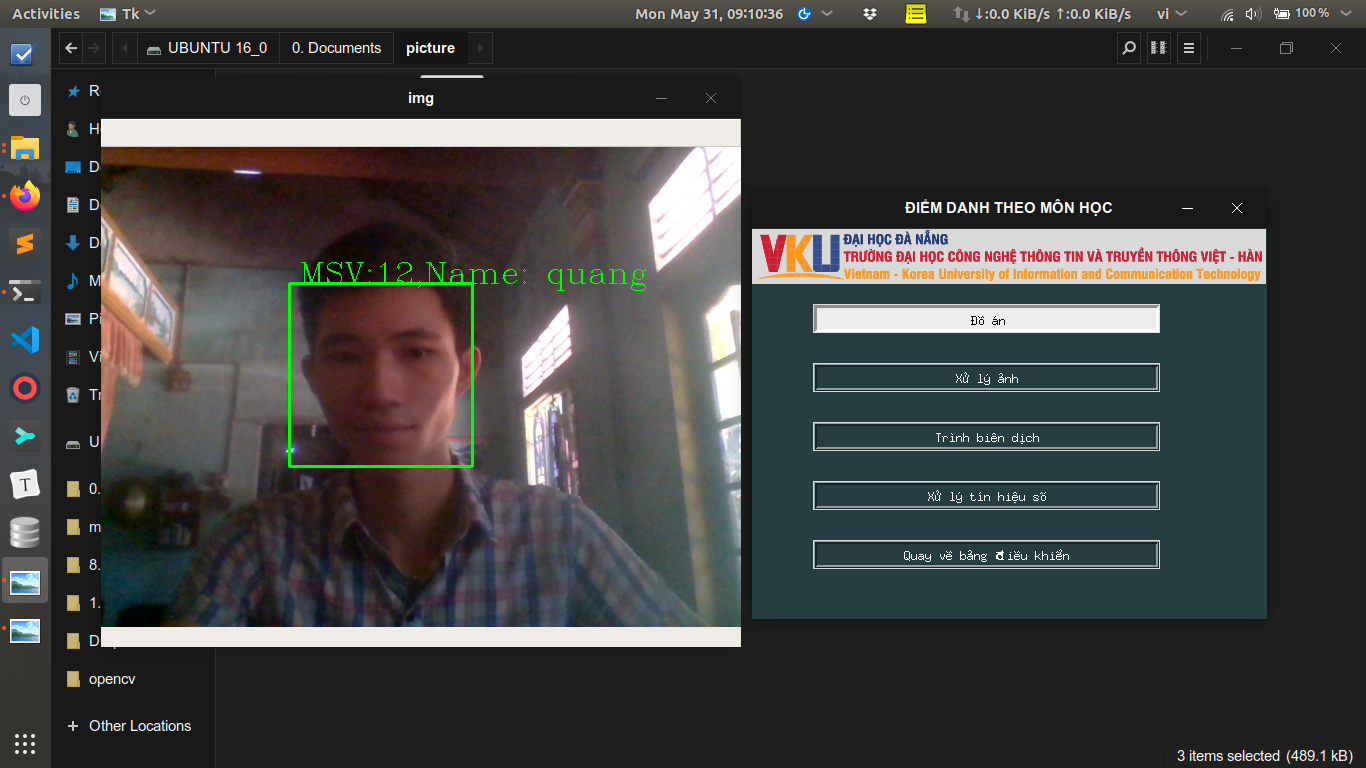
Hình 3.11 Xây dựng mô hình nhận diện

**e) Giao diện bảng điểm danh các môn học**

Giao diện diểm danh chứa danh sách các môn học, và nút lệnh quay về bảng điều khiển. Với mỗi môn học khi nhấn vào thì hệ thống sẽ tiến hành nhận dạng khuôn mặt thông qua camera. Khi nhưng khuôn mặt được phát hiện thì hệ thống sẽ đưa ra mã sinh viên của người đó, thông qua mã sinh viên chúng ta sẽ tìm kiếm trong cơ sỡ dữ liệu rồi hiển thị thông tin người đó lên màn hình và lưu vào danh sách người đã điểm danh.



Hình 3.12 Điểm danh các môn học



Hình 3.13 Thông tin sinh viên khi được nhận dạng

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Đánh giá kết quả

### Kết quả đạt được

Xuyên suốt quá trình thực hiện đồ án từ giai đoạn phân tích đến giai đoạn thực hiện chương trình. Em đã học được thêm nhiều kỹ năng và kiến thức. Kỹ năng tìm kiếm tài liệu, kỹ năng phân tích, giải quyết vấn đề, củng cố kiến thức môn lập trình mạng để ứng dụng vào làm các sản phẩm về sau.

### Hạn chế

* Việc nhận diện các khuôn mặt còn hạn chế một số góc nhìn.
* Thuật toán sử dụng chưa được tối ưu
* Hệ thống chưa có nhiều chức năng nổi trội

## Hướng phát triển

* Cải thiện thuật toán để việc nhận diện được nhanh hơn
* Cải thiện góc nhìn để camera bắt được đầy đủ góc khía cạnh khuôn mặt
* Phát triển thêm nhiều chức năng hơn như điểm danh dựa trên nhận diện khuôn mặt, dự đoán tuổi, dự đoán giới tính.

**DANH MỤC TÀI LIỆU THAM KHẢO**

[1] Asit Kumar Datta, Madhura Datta, Pradipta Kumar Banerjee, Face Detection and Recognition Theory and Practice,2016

[2] Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing, 4 Edition, Pearson Education, 2018

[3] Phùng Thị Thu Hiền, Trần Thị Lan Anh, Giáo trình Trí tuệ nhân tạo, NXB Lao động, 2016