**ĐẠI HỌC ĐÀ NẴNG**

**KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



**ĐỒ ÁN CƠ SỞ 3**

**ĐỀ TÀI:**

**PHẦN MỀM NHẬN DIỆN KHUÔN MẶT**

**VÀ ĐIỂM DANH**

Sinh viên thực hiện : **BÙI XUÂN THIỆN** **18IT3**

**NGUYỄN ĐẶNG KHUÊ VĂN 18IT3**

Giảng viên hướng dẫn : **TS.** **NGUYỄN HÀ HUY CƯỜNG**

Lớp : **18IT3**

***Đà nẵng, tháng 08 năm 2020***

**ĐẠI HỌC ĐÀ NẴNG**

**KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**ĐỒ ÁN CƠ SỞ 3**

**PHẦN MỀM NHẬN DIỆN KHUÔN MẶT**

**VÀ ĐIỂM DANH**

***Đà Nẵng, tháng 08 năm 2020***

**MỞ ĐẦU**

**LỜI CẢM ƠN**

Hiện nay, với sự phát triển của xã hội, vấn đề an ninh bảo mật đang được yêu cầu khắt khe. Hệ thống nhận dạng con người được ra đời với độ tin cậy ngày càng cao. Một trong những bài toán nhận dạng con người rất được quan tâm hiện nay là nhận dạng khuôn mặt. Vì nhận dạng khuôn mặt là cách mà con người sử dụng để phân biệt nhau. Bên cạnh đó, việc thu thập, xử lí thông tin qua ảnh để nhận biết đang được quan tâm và ứng dụng rộng rãi. Với phương pháp này, chúng ta có thể thu thập được nhiều thông tin từ đối tượng mà không cần tác động nhiều đối tượng nghiên cứu. Sự phát triẻn của khoa học máy tính tạo môi trường thuận lợi cho bài toán nhận dạng mặt người từ ảnh số. Việc áp dụng nhận dạng khuôn mặt được sử dụng rộng rãi trong các trường học, cơ quan, văn phòng. Một trong những nhu cầu sử dụng nhận dạng khuôn mặt đó là điểm danh.

Trong đồ án này, nhóm đồ án chúng em sẽ giải quyết vấn đề nhận dạng khuôn mặt để điểm danh theo lớp học có thời gian cụ thể.

Nhóm đồ án chúng em xin gửi lời cảm ơn đến thầy Nguyễn Hà Huy Cường – giảng viên hướng dẫn đồ án cơ sở 3 đã trang bị cho em những kiến thức, kỹ năng cơ bản cần có để hoàn thành đề tài này.

Tuy nhiên trong quá trình nghiêm cứu đề tài, do kiến thức chuyên ngành còn hạn chế nên nhóm chúng em vẫn còn nhiều thiếu sót khi tìm hiểu, đánh giá và trình bày về đề tài. Rất mong nhận được sự quan tấm và góp ý của thầy/cô bộ môn để tài của em được đầy đủ và hoàn chỉnh hơn

Xin chân thành cảm ơn.

**NHẬN XÉT**

(Của cơ quan thực tập, nếu có)

Bold, size 16, xếp sau trang Lời cảm ơn

size 13**………………………………..**

**………………………………………………………………………………………**

**………………………………………………………………………………………**

**………………………………………………………………………………………**

**………………………………………………………………………………………**

**………………………………………………………………………………………**

**NHẬN XÉT**

**(Của giảng viên hướng dẫn)**

**MỤC LỤC**

Trang

[Chương 1 Giới thiệu 1](#_Toc47542779)

[1.1 Tổng quan 1](#_Toc47542780)

[1.2 Mục đích 1](#_Toc47542781)

[1.3 Phương pháp, kết quả 1](#_Toc47542782)

[1.3.1 Phương pháp 1](#_Toc47542783)

[Chương 2 Nghiên cứu tổng quan 2](#_Toc47542784)

[2.1 Các phương pháp 2](#_Toc47542785)

[2.1.1 Bài toán nhận dạng mặt người 2](#_Toc47542786)

[2.1.2 Tổng quan kiến trúc của một hệ thống nhận dạng mặt người 3](#_Toc47542787)

[2.2 Hạn chế, tồn tại của các phương pháp 3](#_Toc47542788)

[2.3 Công nghệ thực hiện 4](#_Toc47542789)

[2.3.1 Giới thiệu ngôn ngữ lập trình Python 4](#_Toc47542790)

[2.3.2 Giới thiệu về thư viện OpenCV 4](#_Toc47542791)

[2.3.3 Môi trường lập trình Visual Studio Code 5](#_Toc47542792)

[2.3.4 Giới thiệu CSDL DB Browser SQLite 6](#_Toc47542793)

[Chương 3 Tổng quan về Nhận Dạng Khuôn Mặt 7](#_Toc47542794)

[3.1 Hệ thống xác định khuôn mặt và ứng dụng 7](#_Toc47542795)

[3.1.1 Định nghĩa 7](#_Toc47542796)

[3.1.2 Một số phương pháp xác định khuôn mặt người 7](#_Toc47542797)

[3.1.3 Phương pháp áp dụng trong đồ án 7](#_Toc47542798)

[Chương 4 Triển khai xây dựng 9](#_Toc47542799)

[4.1 Phân tích thiết kế hệ thống 9](#_Toc47542800)

[4.1.1 Phân tích 9](#_Toc47542801)

[4.1.2 Phân tích thiết kế 9](#_Toc47542802)

[4.2 Cài đặt 10](#_Toc47542803)

[4.3 Viết mã nguồn 10](#_Toc47542804)

[4.3.1 Lấy dữ liệu sinh viên từ bàn phím và webcam 10](#_Toc47542805)

[4.3.2 Traning hình ảnh sinh viên và lưu tr 12](#_Toc47542806)

[4.3.3 Lớp detector nhận diện khuôn mặt đã traning 13](#_Toc47542807)

[Chương 5 Kết luận và Hướng phát triển 15](#_Toc47542808)

[5.1 Kết luận 15](#_Toc47542809)

[5.2 Hướng phát triển 15](#_Toc47542810)

**DANH MỤC HÌNH**

Trang

[Hình 2‑1 Hệ thống nhận dạng khuôn mặt 2](file:///C:\Users\Xuan%20Thien\Desktop\New%20folder\dacs3.docx#_Toc47542669)

[Hình 2‑2 Các bước chính trong hệ thống nhận dạng 3](file:///C:\Users\Xuan%20Thien\Desktop\New%20folder\dacs3.docx#_Toc47542670)

[Hình 2‑3 Ngôn ngữ lập trình Python 4](file:///C:\Users\Xuan%20Thien\Desktop\New%20folder\dacs3.docx#_Toc47542671)

[Hình 2‑4 Giới thiệu về thư viện OpenCV 5](file:///C:\Users\Xuan%20Thien\Desktop\New%20folder\dacs3.docx#_Toc47542672)

[Hình 2‑5 Môi trường lập trình Visual Studio Code 6](file:///C:\Users\Xuan%20Thien\Desktop\New%20folder\dacs3.docx#_Toc47542673)

[Hình 2‑6 Cơ Sở Dữ Liệu SQLite 6](file:///C:\Users\Xuan%20Thien\Desktop\New%20folder\dacs3.docx#_Toc47542674)

[Hình 4‑1 Sơ đồ ngữ cảnh hệ thống 9](file:///C:\Users\Xuan%20Thien\Desktop\New%20folder\dacs3.docx#_Toc47542675)

[Hình 4‑2 Sơ đồ thực hiện chương trình 9](file:///C:\Users\Xuan%20Thien\Desktop\New%20folder\dacs3.docx#_Toc47542676)

[Hình 4‑3 Lấy khuôn mặt từ webcam 11](file:///C:\Users\Xuan%20Thien\Desktop\New%20folder\dacs3.docx#_Toc47542677)

[Hình 4‑4 Lưu thông tin sinh viên vào database 12](file:///C:\Users\Xuan%20Thien\Desktop\New%20folder\dacs3.docx#_Toc47542678)

[Hình 4‑5 Kết quả chương trình nhận dạng 14](file:///C:\Users\Xuan%20Thien\Desktop\New%20folder\dacs3.docx#_Toc47542679)

**DANH MỤC CỤM TỪ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| **STT** | **Cụm từ** | **Viết tắt** |
| 1 | Cơ Sở Dữ Liệu | CSDL |
| 2 | Công nghệ thông tin | CNTT |

# Giới thiệu

## Tổng quan

Phổ biến trong đời sống con người đã làm cho lượng thông tin thu được bằng hình ảnh ngày càng tăng. Theo đó, lĩnh vực xử lý ảnh cũng được chú trọng phát triển, ứng dụng rộng rãi trong đời sống xã hội hiện đại. Không chỉ dừng lại ở việc chỉnh sửa, tăng chất lượng hình ảnh mà với công nghệ xử lý ảnh hiện nay chúng ta có thể giải quyết các bài toán nhận dạng chữ viết, nhận dạng dấu vân tay, nhận dạng khuôn mặt…

Một trong những bài toán được nhiều người quan tâm nhất của lĩnh vực xử lý ảnh hiện nay đó là nhận dạng khuôn mặt (Face Recognition). Như chúng ta đã biết, khuôn mặt đóng vai trò quan trọng trong quá trình giao tiếp giữa người với người, nó mang một lượng thông tin giàu có, chẳng hạn như từ khuôn mặt chúng ta có thể xác định giới tính, tuổi tác, chủng tộc, trạng thái cảm xúc, đặc biệt là xác định mối quan hệ với đối tượng (có quen biết hay không). Do đó, bài toán nhận dạng khuôn mặt đóng vai trò quan trọng trong nhiều lĩnh vực đời sống hằng ngày của con người như các hệ thống giám sát, quản lý vào ra, tìm kiếm thông tin một người nổi tiếng,…đặc biệt là an ninh, bảo mật. 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 nghiêng, kích thước ảnh, hay ảnh hưởng của tham số môi trường.

Có hai phương pháp nhận dạng phổ biến hiện nay là nhận dạng dựa trên đặc trưng của các phần tử trên khuôn mặt như biến đổi Gabor Wavelet và mạng Neural, SVM,…và nhận dạng dựa trên xét tổng thể toàn khuôn mặt như phương pháp PCA, LDA, LFA .

Chính nhờ những yếu tố trên, nhóm chúng em đã đưa ra ý tưởng tạo ra phần mềm **nhận diện khuôn mặt và điểm danh**. Được dựa trên thư viện OpenCV.

## Mục đích

- Xây dựng một dự án phần mềm ứng dụng nhận dạng khuôn mặt.

- Tìm hiểu về thư viện OpenCv.

- Tìm hiểu các phương pháp xác định khuôn mặt (Face Detection)

## Phương pháp, kết quả

### Phương pháp

a) Đối tượng:

- Các phương pháp, thuật toán phục vụ cho việc phát hiện và nhận dạng khuôn mặt người trên ảnh.

- Bộ thư viện xử lý ảnh OpenCv

- CSDL tự tạo bằng cách thu thập dữ liệu sinh viên(khuôn mặt)

b) Phạm vi nghiên cứu:

- Tập trung tìm hiểu nhận dạng khuôn mặt (Face Recognition)

- Xử lí ảnh thô đưa vào database.

- Nhận dạng khuôn mặt và điểm danh .

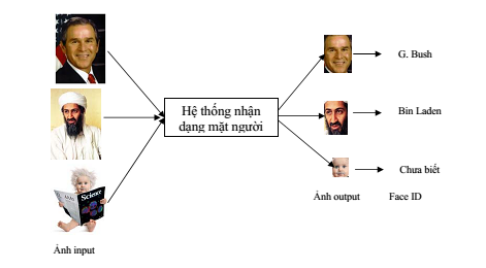
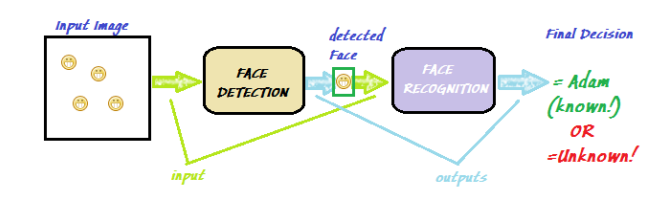
# Nghiên cứu tổng quan

## Các phương pháp

### Bài toán nhận dạng mặt người

Hệ thống nhận dạng mặt người là một hệ thống nhận vào là một ảnh,một đoạn video (một dòng các hình ảnh liên tục) hoặc bằng webcam. Qua xử lý, tính toán hệ thống xác định được vị trí mặt người (nếu có) trong ảnh và xác định là người nào trong số những người mà hệ thống đã được biết (qua quá trình học) hoặc là người lạ.

Trước khi nhận dạng mặt người, chương trình phải chuẩn bị trước một dữ liệu thô. Bao gồm các khuôn mặt của người cụ thể là sinh viên. Tiếp theo là training dữ liệu thô để trích xuất các embedding vector để phục vụ cho việc so sánh sau này. Do đó việc chuẩn bị dữ liệu ban đầu hết sức quan trọng.



Hình ‑ Hệ thống nhận dạng khuôn mặt

### Tổng quan kiến trúc của một hệ thống nhận dạng mặt người

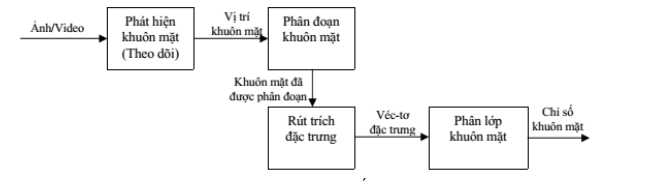
Một hệ thống nhận dạng mặt người thông thường bao gồm bốn bước xử lý sau:

1. Phát hiện khuôn mặt (Face Detection).

2. Phân đoạn khuôn mặt (Face Alignment hay Segmentation).

3. Trích chọn đặc trưng (Feature Extraction).

4. Nhận dạng (Recognition) hay Phân lớp khuôn mặt (Face Clasaification).



Hình ‑ Các bước chính trong hệ thống nhận dạng

## Hạn chế, tồn tại của các phương pháp

Bài toán nhận dạng mặt người là bài toán đã được nghiên cứu từ những năm 70. Tuy nhiên, đây là một bài toán khó nên những nghiên cứu hiện tại vẫn chưa đạt được những kết quả mong muốn. Chính vì thế, vấn đề này vẫn đang được nhiều nhóm trên thế giới quan tâm nghiên cứu. Khó khăn của bài toán nhận dạng mặt người có thể kể đến như sau:

a. Tư thế chụp, góc chụp: Ảnh chụp khuôn mặt có thể thay đổi rất nhiều bởi vì góc chụp giữa camera và khuôn mặt. Chẳng hạn như: chụp thẳng, chụp chéo bên trái 45o hay chụp chéo bên phải 45o , chụp từ trên xuống, chụp từ dưới lên, v.v… Với các tư thế khác nhau, các thành phần trên khuôn mặt như mắt, mũi, miệng có thể bị khuất một phần hoặc thậm chí khuất hết.

b. Sự xuất hiện hoặc thiếu một số thành phần của khuôn mặt: Các đặc trưng như: râu mép, râu hàm, mắt kính, v.v… có thể xuất hiện hoặc không. Vấn đề này làm cho bài toán càng trở nên khó hơn rất nhiều.

c. Sự biểu cảm của khuôn mặt: Biểu cảm của khuôn mặt con người có thể làm ảnh hưởng đáng kể lên các thông số của khuôn mặt. Chẳng hạn, cùng một khuôn mặt một người, nhưng có thể sẽ rất khác khi họ cười hoặc sợ hãi, v.v…

d. Sự che khuất: Khuôn mặt có thể bị che khuất bởi các đối tượng khác hoặc các khuôn mặt khác.

e. Hướng của ảnh (pose variations): Các ảnh khuôn mặt có thể biến đổi rất nhiều với các góc quay khác nhau của trục camera. Chẳng hạn chụp với trục máy ảnh nghiêng làm cho khuôn mặt bị nghiêng so với trục của ảnh.

f. Điều kiện của ảnh: Ảnh được chụp trong các điều kiện khác nhau về: chiếu sáng, về tính chất camera (máy kỹ thuật số, máy hồng ngoại, v.v…), ảnh có chất lượng thấp ảnh hưởng rất nhiều đến chất lượng ảnh khuôn mặt. g. Aging condition: Việc nhận dạng ảnh mặt thay đổi theo thời gian còn là một vấn đề khó khăn, ngay cả đối với khả năng nhận dạng của con người.

## Công nghệ thực hiện

Để giải quyết bài toán nhận dạng khuôn mặt sử dụng thư viện OpenCV, chúng ta có thể sử dụng các ngôn ngữ lập trình như: .NET C#, VB, IronPython, Java, C++…

Trong đồ án này ngôn ngữ lập trình được sử dụng là Python, được viết trên môi trường của Python là IDLE và Visual Studio Code.

### Giới thiệu ngôn ngữ lập trình Python

Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu.

Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động; do vậy nó tương tự như Perl, Ruby, Scheme, Smalltalk, và Tcl. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý.

Ban đầu, Python được phát triển để chạy trên nền Unix. Nhưng rồi theo thời gian, Python dần mở rộng sang mọi hệ điều hành từ MS-DOS đến Mac OS, OS/2, Windows, Linux và các hệ điều hành khác thuộc họ Unix. Mặc dù sự phát triển của Python có sự đóng góp của rất nhiều cá nhân, nhưng Guido van Rossum hiện nay vẫn là tác giả chủ yếu của Python. Ông giữ vai trò chủ chốt trong việc quyết định hướng phát triển của Python.



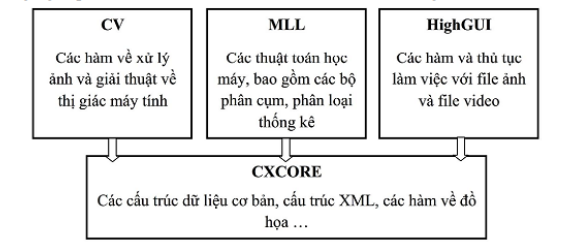
Hình ‑ Ngôn ngữ lập trình Python

### Giới thiệu về thư viện OpenCV

OpenCv (Open Source Computer Vision) là một thư viện mã nguồn mở về thị giác máy với hơn 500 hàm và hơn 2500 các thuật toán đã được tối ưu về XLA, và các vấn đề liên quan tới thị giác máy. OpenCv được thiết kế một cách tối ưu, sử dụng tối đa mạnh của các dòng chip đa lõi… để thực hiện các phép tính toán trong thời gian thực, nghĩa là tốc độ đáp ứng của nó thể đủ nhanh cho các ứng dụng thông thường.

OpenCv là thư viện được thiết kế để chạy trên nhiều nền tảng khác nhau (cross-platform),nghĩa là nó có thể chạy trên hệ điều hành Window, Linux, Mac, iOS… Việc sử dụng thư viện OpenCv tuân theo các quy định về sử dụng phần mềm mã nguồn mở BSD do đó bạn có thể sử dụng thư viện này một cách miễn phí cho các mục đích phi thương mại lẫn thương mại.

Dự án về OpenCv được khởi động từ những năm 1999, đến năm 2000 nó được giới thiệu trong một hội nghị của IEEE về các vấn đề trong thị giác máy và nhận dạng, tuy nhiên bản OpenCV 1.0 mãi tới tận năm 2006 mới chính thức được công bố và năm 2008 bản 1.1 (prerelease) mới được ra đời. Tháng 10 năm 2009, bản OpenCV thế hệ thứ hai ra đời (thường gọi là phiên bản 2.x), phiên bản này có giao diện của C++ (khác với phiên bản trước có giao diện của C) và có nhiều điểm khác biệt so với phiên bản thứ nhất.

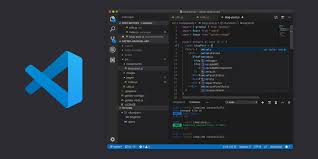
Thư viện OpenCV ban đầu được sự hỗ trợ từ Intel, sau đó được hỗ trợ bởi Willow Garage, một phòng thí nghiệm chuyên nghiên cứu về công nghệ robot. Cho đến nay, OpenCV vẫn là thư viện mở, được phát triển bởi nguồn quỹ không lợi nhuận (none-profit foundation) và được sự hưởng ứng rất lớn của cộng đồng.

Hình ‑ Giới thiệu về thư viện OpenCV

### Môi trường lập trình Visual Studio Code

Visual Studio Code là một trình biên tập mã được 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. Nó cũng cho phép tùy chỉnh, do đó, người dùng có thể thay đổi theme, phím tắt, và cá tùy chọn khác. Nó miễn phí và là phần mềm mã nguồn mở, mặc dù gói tải xuống chính thì là có giấy phép.

Visual Studio Code được dựa trên Electron, một nền tảng được sử dụng để triển khai các ứng dụng Node.js máy tính cá nhân chạy trên động cơ bố trí Blink. Mặc dù nó sử dụng nền tảng Electron nhưng phần mềm này không phải là một bản khác của Atom, nó thực ra được dựa trên trình biên tập của Visual Studio Online (tên mã là "Monaco")



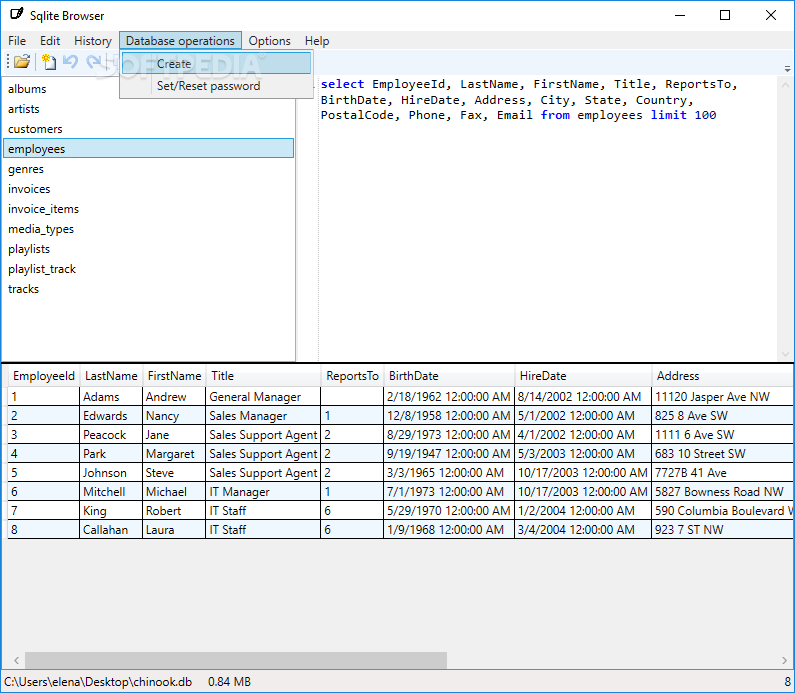
Hình ‑ Môi trường lập trình Visual Studio Code

### Giới thiệu CSDL DB Browser 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 là một lựa chọn phổ biến vì phần mềm cơ sở dữ liệu nhúng để lưu trữ cục bộ / máy khách trong phần mềm ứng dụng như trình duyệt web . Nó được cho là công cụ cơ sở dữ liệu được triển khai rộng rãi nhất , vì ngày nay nó được sử dụng bởi một số trình duyệt, hệ điều hành và hệ thống nhúng rộng rãi (như điện thoại di động). SQLite có các ràng buộc với nhiều ngôn ngữ lập trình.

SQLite lưu trữ toàn bộ cơ sở dữ liệu (định nghĩa, bảng, chỉ mục và chính dữ liệu) dưới dạng một tệp đa nền tảng trên một máy chủ. Nó thực hiện thiết kế đơn giản này bằng cách khóa toàn bộ tệp cơ sở dữ liệu trong khi viết. Các hoạt động đọc SQLite có thể được đa nhiệm, mặc dù việc ghi chỉ có thể được thực hiện tuần tự.

Mục tiêu thiết kế của SQLite là cho phép chương trình được vận hành mà không cần cài đặt hệ thống quản lý cơ sở dữ liệu hoặc yêu cầu người quản trị cơ sở dữ liệu . Hipp dựa trên cú pháp và ngữ nghĩa trên PostgreSQL 6.5. Vào tháng 8 năm 2000, phiên bản 1.0 của SQLite đã được phát hành, với lưu trữ dựa trên gdbm (Trình quản lý cơ sở dữ liệu GNU). SQLite 2.0 đã thay thế gdbm bằng cách triển khai cây B tùy chỉnh , thêm khả năng giao dịch . SQLite 3.0, được tài trợ một phần bởi America Online , đã thêm quốc tế hóa , gõ bảng kê khai và các cải tiến lớn khác.

Hình ‑ Cơ Sở Dữ Liệu SQLite

# Tổng quan về Nhận Dạng Khuôn Mặt

## Hệ thống xác định khuôn mặt và ứng dụng

### Định nghĩa

Xác định khuôn mặt người (Face Detection) là một kỹ thuật máy tính để xác định các vị trí và các kích thước của các 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…

### Một số phương pháp xác định khuôn mặt người

Trên thực tế người ta hay chia các phương pháp nhận dạng mặt ra làm 3 loại: phương pháp tiếp cận toàn cục (global, như Eigenfaces-PCA, Fisherfaces-LDA), phương pháp tiếp cận dựa trên các đặc điểm cục bộ (local feature based, như LBP, Gabor wavelets) và phương pháp lai (hybrid, là sự kết hợp của hai phương pháp toàn cục và local feature).

Phương pháp dựa trên các đặc điểm cục bộ đã được chứng minh là ưu việt hơn khi làm việc trong các điều kiện không có kiểm soát và có thể nói rằng lịch sử phát triển của nhận dạng mặt (A never ending story) là sự phát triển của các phương pháp trích chọn đặc trưng (feature extractrion methods) được sử dụng trong các hệ thống dựa trên feature based.

Các ứng dụng cụ thể của nhận dạng mặt dựa trên 2 mô hình nhận dạng: identification (xác định danh tính, bài toán 1-N), và verification (xác thực danh tính, bài toán 1-1). Trong bài toán identification, ta cần xác định danh tính của ảnh kiểm tra, còn trong bài toán verification ta cần xác định 2 ảnh có cùng thuộc về một người hay không.

### Phương pháp áp dụng trong đồ án

Để xây dựng một hệ thống nhận dạng mặt, cũng không hề đơn giản, bước đầu tiên cần thực hiện là face detection, tức là phát hiện phần ảnh mặt trong dữ liệu input (CSDL ảnh, video …) và cắt lấy phần ảnh mặt để thực hiện nhận dạng (face cropping), bước thứ hai là tiền xử lý ảnh (preprocessing) bao gồm các bước căn chỉnh ảnh (face image alignment) và chuẩn hóa ánh sáng (illumination normalization), tiếp đến là bước trích chọn đặc điểm (feature extraction), ở bước này một phương pháp trích chọn đặc điểm nào đó (mẫu nhị phân cục bộ – Local Binary Pattern – LBP, Gabor wavelets, …) sẽ được sử dụng với ảnh mặt để trích xuất các thông tin đặc trưng cho ảnh, kết quả là mỗi ảnh sẽ được biểu diễn dưới dạng một vector đặc điểm (feature vector), bước tiếp theo là bước nhận dạng (recognition) hay phân lớp (classification), tức là xác định danh tính (identity) hay nhãn (label) của ảnh – đó là ảnh của ai.

Ở bước classification, thường thì phương pháp k-láng giềng gần nhất (k-nearest neighbor:kNN) sẽ được sử dụng, thực tế cho thấy việc dùng SVM (Support Vector Machine) không mang lại hiệu quả cao hơn mà còn chậm hơn.

Dữ liệu cho một hệ thống nhận dạng mặt được chia làm 3 tập: tập huấn luyện (training set), tập tham chiếu (reference set hay gallery set) và tập để nhận dạng (probe set hay query set, đôi khi còn gọi là test set). Trong nhiều hệ thống, tập training trùng với tập reference.

Tập training gồm các ảnh được dùng để huấn luyện (hay học-learning), thông thường tập này được dùng để sinh ra một không gian con (projection subspace) là một ma trận và phương pháp hay được sử dụng là PCA (Principal Component Analysis), WPCA (Whitened PCA), LDA (Linear Discriminant Analysis), KPCA (Kernel PCA). Tập reference gồm các ảnh đã biết danh tính được chiếu (projected) vào không gian con ở bước training.

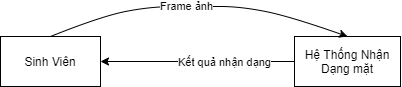
Bước training nhằm 2 mục đích: giảm số chiều (dimension reduction) của các vector đặc điểm (feature vector) vì các vector này thường có độ dài khá lớn (vài nghìn tới vài trăm nghìn) nên nếu để nguyên thì việc tính toán sẽ rất rất lâu, thứ hai là làm tăng tính phân biệt (discriminative) giữa các ảnh khác lớp (định danh khác nhau), ngoài ra có thể làm giảm tính phân biệt giữa các ảnh thuộc về một lớp (tùy theo phương pháp, ví dụ như Linear Discriminant Analysis LDA- còn gọi là Fisher Linear Discriminant Analysis-Fisherface là một phương pháp làm việc với tập training mà mỗi đối tượng có nhiều ảnh mặt ở các điều kiện khác nhau).

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). Nhận dạng (hay phân lớp) được thực hiện với tập các ảnh probe, sau khi tiền xử lý xong, mỗi ảnh sẽ được áp dụng phương pháp trích chọn đặc điểm (như với các ảnh thuộc tập training và reference) và được chiếu vào không gian con. Tiếp đến việc phân lớp sẽ dựa trên phương pháp k-NN, định danh của một ảnh cần xác định sẽ được gán là định danh của ảnh có khoảng cách (distance) gần với nó nhất. Ở đây cần lưu ý là mỗi ảnh là một vector nên có thể dùng khái niệm hàm khoảng cách giữa hai vector để đo sự khác biệt giữa các ảnh.

# Triển khai xây dựng

## Phân tích thiết kế hệ thống

### Phân tích

Nhiệm vụ chính của chương trình là nhận dạng một khuôn mặt người xem khuôn mặt đó có được biết đến hay chưa. Ngoài ra, chương trình còn thực hiện công việc phát hiện, tách các khuôn mặt người (nếu có) từ một ảnh tĩnh, hoặc từ các frame ảnh thu được từ camera. Sau đó, lưu vào CSDL làm tập mẫu.

Hình ‑ Sơ đồ ngữ cảnh hệ thống

- Như vậy, các chức năng của chương trình bao gồm:

o Lấy một ảnh từ kết nối đến webcam hiển thị lên

o Thực hiện tách các khuôn mặt

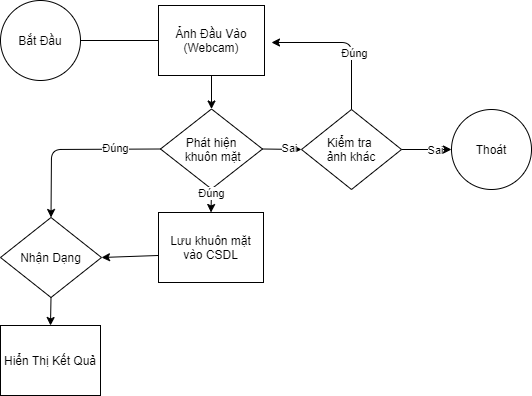
o Thực hiện lưu khuôn mặt phát hiện được vào CSDL

o Nhận dạng ảnh một khuôn mặt. Nếu “biết” người đó (có lưu thông tin trong CSDL) thì hiển thị tên và một ảnh của người đó. Nếu “không biết” (không có thông tin của người đó trong CSDL) thì hiển thị 1 màu đen lên màn hình và tên người là: “Unknow”

### Phân tích thiết kế

Với các chức năng nêu trên, chương trình được chia thành 5 phần chính:

* Xử lý ảnh đầu vào.
* Phát hiện khuôn mặt.
* Xử lí khuôn mặt
* Xử lý đầu ra
* Lưu điểm danh



Hình ‑ Sơ đồ thực hiện chương trình

## Cài đặt

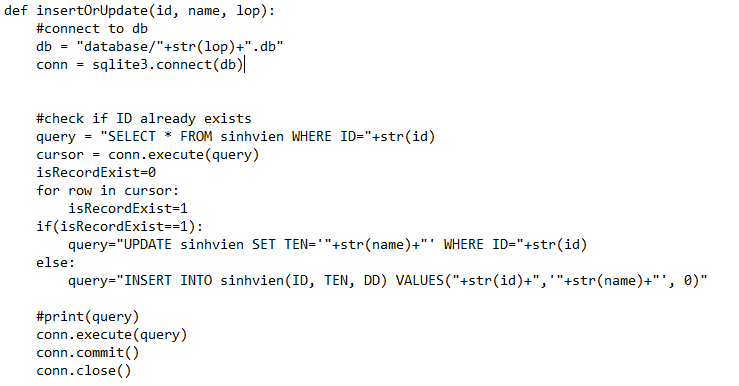
* Cài đặt Python 2.7 và OpenCV
* Cài đặt numpy
  + Vào thư mục \Python27\Scripts chạy lệnh
* Cài đặt pilow
  + Vào thư mục \Python27\Scripts chạy lệnh
* Cài đặt sqlite
  + Tạo CSDL FaceBase với bảng sinhvien

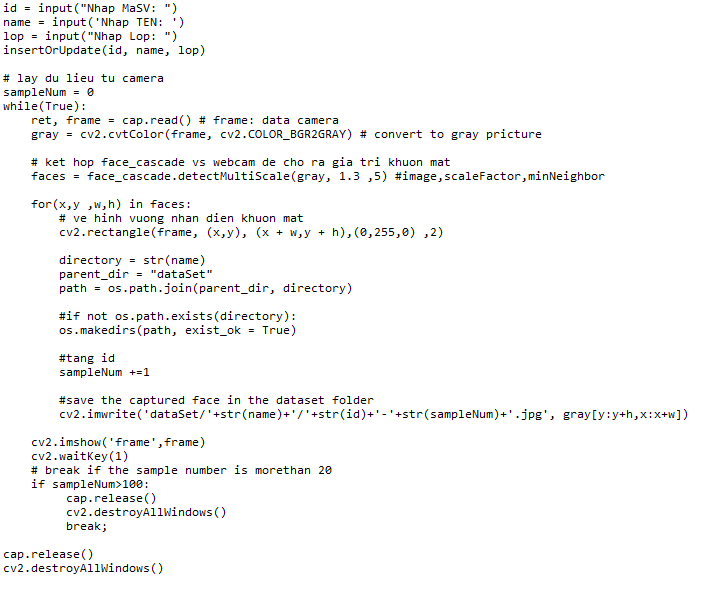
## Viết mã nguồn

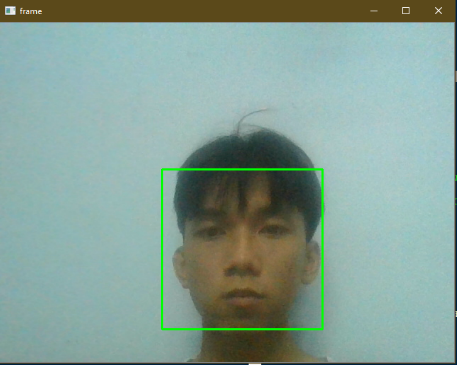
* Tạo dataSet dùng để nhận diện
* Tạo dataTraning sử dụng để traning dữ liệu
* Tạo lớp detector để nhận diện

### Lấy dữ liệu sinh viên từ bàn phím và webcam

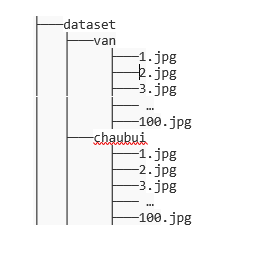
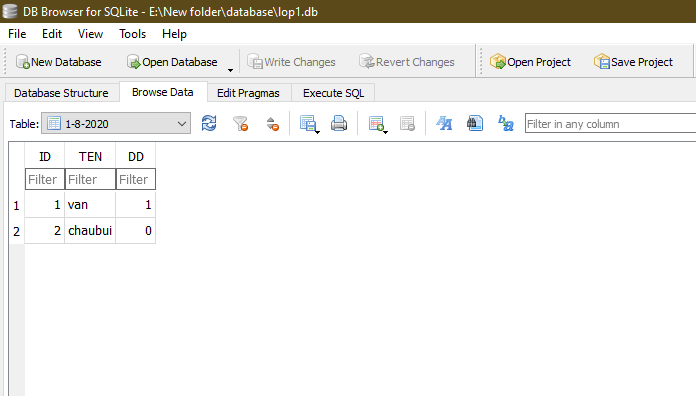
* Kết nối sqlite để tạo/cập nhật record



*  Dùng camera để chụp hình dữ liệu, nhập thông tin sinh viên
* Dữ liệu ban đầu được cài đặt giới hạn 100 tấm hình



Hình ‑ Lấy khuôn mặt từ webcam

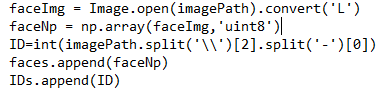
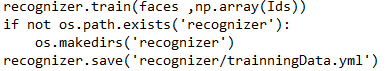
*  Thông tin sinh viên sẽ được lưu vào database và hình ảnh sẽ được đưa vào file dataset với folder là tên sinh viên

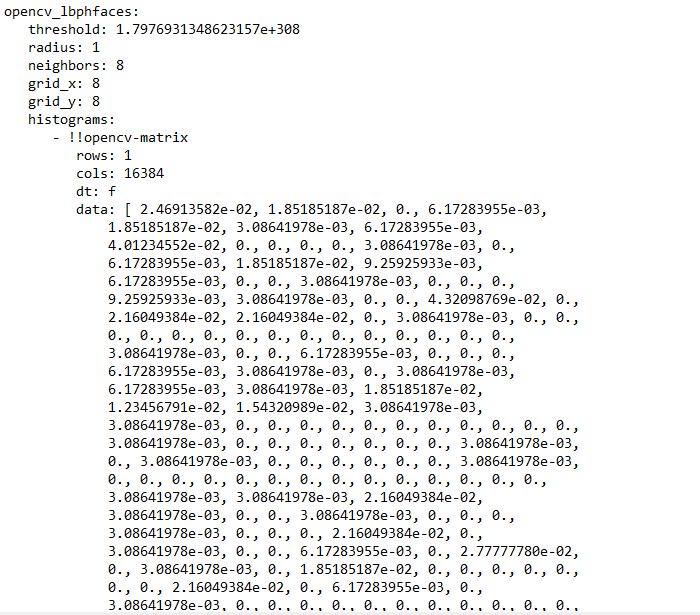
Hình ‑ Lưu thông tin sinh viên vào database

### Traning hình ảnh sinh viên và lưu trữ

* Load đường dẫn và file trong thư mục dataset

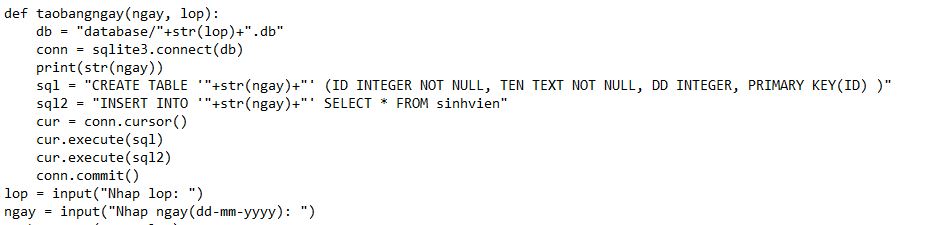


*  Tách ID từ tên ảnh, add vào mảng faces vùng ID
*  Traning ảnh và lưu vào file

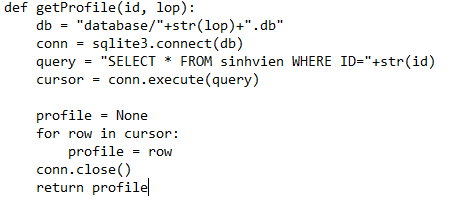


### Lớp detector nhận diện khuôn mặt đã traning

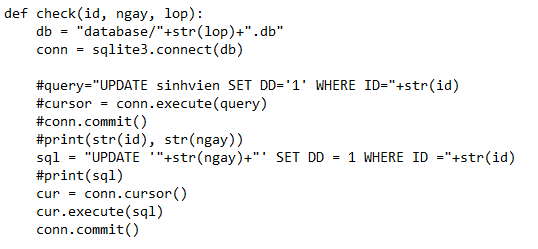
* Tạo thông tin ngày và lớp điểm danh để bắt đầu điểm danh



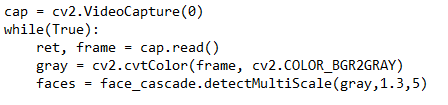
* Lấy thông tin sinh viên từ database



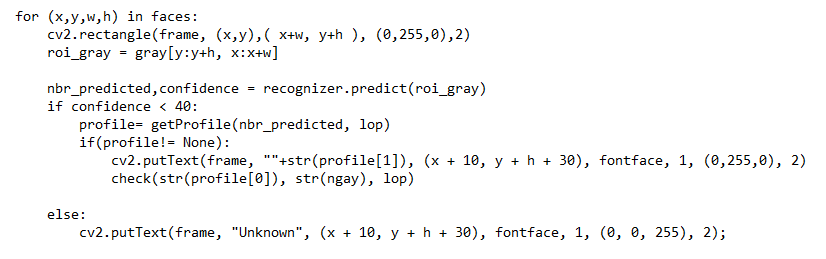
* Cập nhập điểm danh sinh viên trong database

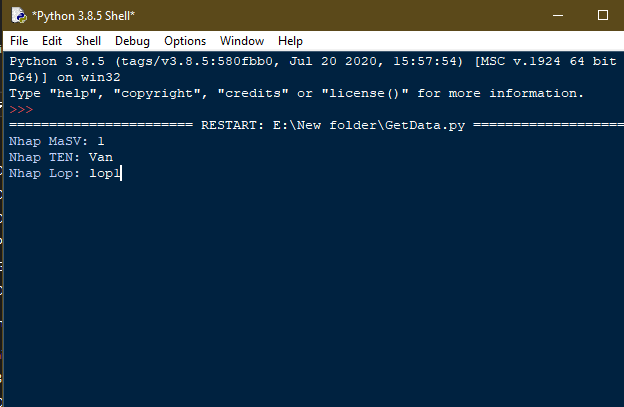


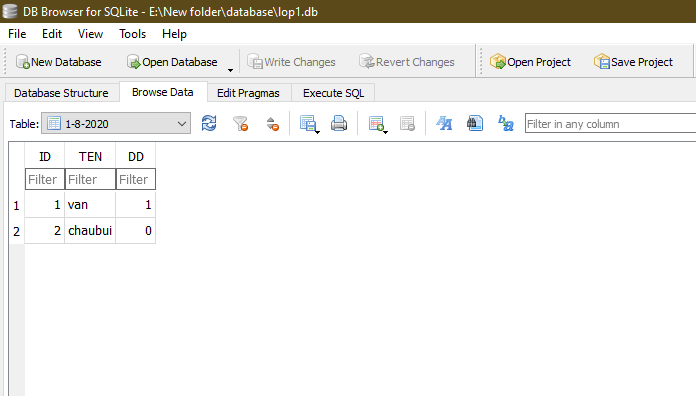
* Sử dụng camera để nhận diện



* Hiện thị thông tin sinh viên lên màn hình



*  Thoát màn hình điểm danh
*  Kết quả:



Hình ‑ Kết quả chương trình nhận dạng

# Kết luận và Hướng phát triển

## Kết luận

Báo cáo đồ án đã trình bày được kiến thức cơ bản để giải quyết bài toán nhận dạng khuôn mặt người. Riêng phương pháp nhận dạng mặt người bằng thư viện OpenCV được trình bày ở chương 2 và chương 3. Từ kết quả cho thấy, độ chính xác của chương trình chỉ đạt ở 70% nhận dạng đúng. Tuy nhiên, do thời gian hạn chế nên chương trình còn nhiều khuyết điểm và nhiều ý tưởng chưa được thực hiện.

Để chương trình có thể sử dụng trong thực tế cần rất nhiều thời gian nghiên cứu và giải quyết các vấn đề gặp phải, thực hiện các ý tưởng mới để nâng cao tốc độ, hiệu suất và độ chính xác của chương trình.

Các ý tưởng mới:

- Thực hiện điểm danh bằng camera giám sát của lớp học. Không cần phải thực hiện chỉnh tay như thông thường. Vì khi dùng webcam điểm danh lần lượt có nhiều hạn chế nhất định.

-

## Hướng phát triển

Trong tương lai, phần mềm nhận dạng khuôn mặt và điểm danh có thể phát triển vượt bậc, dùng thuật toán mới hơn và xử lí ảnh tốt hơn để nhận dạng khuôn mặt một cách chính xác. Có thể giám sát chủ động theo thời gian thực tất cả các đối tượng trong phạm vi.

Áp dụng giải quyết các bài toán kẹt xe bằng cách giám sát, kiểm soát lưu lượng lưu thông và mật độ xe và từ đó đưa ra các cảnh báo và hướng dẫn. Các hệ thống nhận dạng khuôn mặt hiện đang được nghiên cứu hoặc triển khai cho bộ phận an ninh sân bay. Dữ liệu từ một hệ thống nhận dạng khuôn mặt có thể được thu thập và lưu trữ mà đối tượng không hay biết. Thông tin sau đó có thể bị tin tặc hoặc kẻ nào đó truy cập với mục đích xấu.

**DANH MỤC TÀI LIỆU THAM KHẢO**

1. [Nhận diện khuôn mặt bằng python và OpenCV](https://www.youtube.com/watch?v=o7YaiI42jZg&lc=Ugy6Pi8TuJ1RI6VFG2B4AaABAg.9B7eQDYRuYS9B8AE7mlla-)
2. [Nhận diện khuôn mặt với Python và OpenCV](https://blog.vietnamlab.vn/2018/06/29/nhan-dien-khuon-mat-voi-python-va-opencv/#2vitmngun) phần 2
3. [Kỹ thuật nhận dạng face Recognition](https://techblog.vn/opencv-nhan-dang-face-recognition)
4. [Thị giác máy tính](https://thigiacmaytinh.com/nhan-dien-khuon-mat-p1-face-detection/)