# **ĐẠI HỌC ĐÀ NẴNG**

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



**BÁO CÁO ĐỒ ÁN CƠ SỞ 5**

**ĐỀ TÀI:**

**CHƯƠNG TRÌNH XỬ LÍ ẢNH NHẬN DIỆN DÙNG TRONG CHẤM CÔNG, ĐIỂM DANH**

Sinh viên thực hiện: **TRẦN ĐÌNH NGHĨA 17IT020**

**TRẦN MINH THÍCH 17IT028**

Lớp : **17IT1**

Giảng viên hướng dẫn: **TS. LÊ VĂN MINH**

*Đà Nẵng, tháng 6 năm 2020*

# **Lời cảm ơn**

Để hoàn thành chuyên đề báo cáo đồ án cơ sở này trước tiên chúng em xin gửi đến các quý thầy, cô giáo Khoa CNTT & TT Đại Học Đà Nẵng lời cảm ơn chân thành và sâu sắc nhất.

Đặc biệt, chúng em xin gởi đến thầy Lê Văn Minh – người đã tận tình hướng dẫn, giúp đỡ chúng em hoàn thành chuyên đề đồ án cơ sở 5 này lời cảm ơn sâu sắc nhất.

Vì kiến thức bản thân còn hạn chế, trong quá trình thực hiện đồ án chúng em không tránh khỏi những sai sót, kính mong nhận được những ý kiến đóng góp từ quý thầy cô

Chúng em xin chân thành cảm ơn.

# **Mở đầu**

Về công nghệ nhận dạng khuôn mặt, đây là ứng dụng dựa vào các đặc điểm trên khuôn mặt để so sánh, nhận dạng và định danh 1 đối tượng 1 cách tự động thông qua cơ sở dữ liệu khuôn mặt đã được lưu trữ từ trước. Đây không phải là công nghệ mới, trên thực tế thì từ khi công nghệ sinh trắc học phát triển và bùng nổ thì công nghệ nhận diện khuôn mặt cũng được phát triển theo. So với nhận diện bằng dấu vân tay hay ID thì nhận diện bằng khuôn mặt tiết kiệm được thời gian hơn rất nhiều.

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

**…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………**

Chữ kí giảng viên hướng dẫn

**MỤC LỤC**

[**Chương 1: Giới thiệu** 5](file:///C:\Users\PC\Desktop\Đồ%20Án%205.docx#_Toc47438540)

[**1.1** **Giới thiệu** 6](#_Toc47438541)

[**1.2** **Mục đích và mục tiêu dự án** 6](#_Toc47438542)

[**1.3** **Lưu đồ thuật toán** 7](#_Toc47438543)

[**Chương 2: Xem xét tài liệu** 9](file:///C:\Users\PC\Desktop\Đồ%20Án%205.docx#_Toc47438544)

[**2.1 Xử lí ảnh kỹ thuật số** 10](#_Toc47438545)

[**2.2 Đại diện hình ảnh trong máy tính kỹ thuật số** 10](#_Toc47438546)

[**2.3 Các bước xử lí hình ảnh kỹ thuật số** 11](#_Toc47438547)

[**2.4 Định nghĩa** 11](#_Toc47438548)

[**2.4.1 Phát hiện khuôn mặt** 11](#_Toc47438549)

[**2.4.2 Nhận diện khuôn mặt** 12](#_Toc47438550)

[**2.4.3 Phân biệt giữa “Phát hiện khuôn mặt” và “Nhận diện khuôn mặt”** 12](#_Toc47438551)

[**2.4.4 Thuật toán phát hiện khuôn mặt LBPH ( Local Binary Patterns Histogram )** 12](#_Toc47438552)

[**Chương 3: Mô hình thực hiện và phân tích** 18](file:///C:\Users\PC\Desktop\Đồ%20Án%205.docx#_Toc47438553)

[**3.1 Mô hình thực hiện** 19](#_Toc47438554)

[**3.1.1 Phần mềm triển khai** 19](#_Toc47438555)

[**3.1.2 Phần cứng triểu khai** 19](#_Toc47438556)

[**3.2 Kết quả thực nghiệm** 20](#_Toc47438557)

[**Chương 4: Tiến hành lập trình** 22](#_Toc47438558)

[**4.1 Tiến hành lập trình** 23](#_Toc47438559)

[**4.1.1 haarcascade\_frontalface\_defaultx.xml** 23](#_Toc47438560)

[**4.1.2 main.py** 24](#_Toc47438561)

[**4.1.3 Check\_camera.py** 26](#_Toc47438562)

[**4.1.4 Capture\_Image.py** 26](#_Toc47438563)

[**4.1.5 Train\_Image.py** 28](#_Toc47438564)

[**4.1.6 Recognize.py** 28](#_Toc47438565)

[**Chương 5: Tổng kết** 30](#_Toc47438566)

[**5.1 Hình ảnh kết quả chương trình** 31](#_Toc47438567)

[**5.2 Ưu điểm và hạn chế của dự án** 35](#_Toc47438568)

[**5.3 Hướng phát triển** 36](#_Toc47438571)

[**Tài liệu tham khảo:** 37](#_Toc47438572)

# **Chương 1: Giới thiệu**

* 1. **Giới thiệu**

Điểm danh hay chấm công là một điều quan trọng đối với doanh nghiệp, tổ chức giáo dục. Vì vậy việc giữ hồ sơ thông tin tham dự, có mặt cũng rất quan trọng. Vấn đề phát sinh khi nói về quy trình truyền thống, gọi tên học sinh, sinh viên để điểm danh không chỉ tốn thời gian mà còn cả năng lượng.

Vì thế, một hệ thống điểm danh tự động có thể giải quyết vấn đề trên. Trong dự án này, nhận diện khuôn mặt được được sử dụng dựa trên xử lí ảnh. Chương trình sẽ xác định vị trí vùng mặt và nhận dạng khuôn mặt để đánh dấu sự có mặt, tham dự của nhân viên, học sinh, sinh viên. Cơ sở dữ liệu của tất cả mọi người được lưu trữ và khi khuôn mặt đó khớp với một trong các khuôn mặt được lưu trữ trong cơ sở dữ liệu thì sự có mặt, tham dự được ghi lại. Ngoài ra, các học sinh, sinh viên hay nhân viên phải đăng kí vào cơ sở dữ liệu để được nhận diện.

* 1. **Mục đích và mục tiêu dự án**

Mục tiêu của dự án này là phát triển nhận diện khuôn mặt hệ thống điểm danh. Thành tựu mong đợi để thực hiện mục tiêu là:

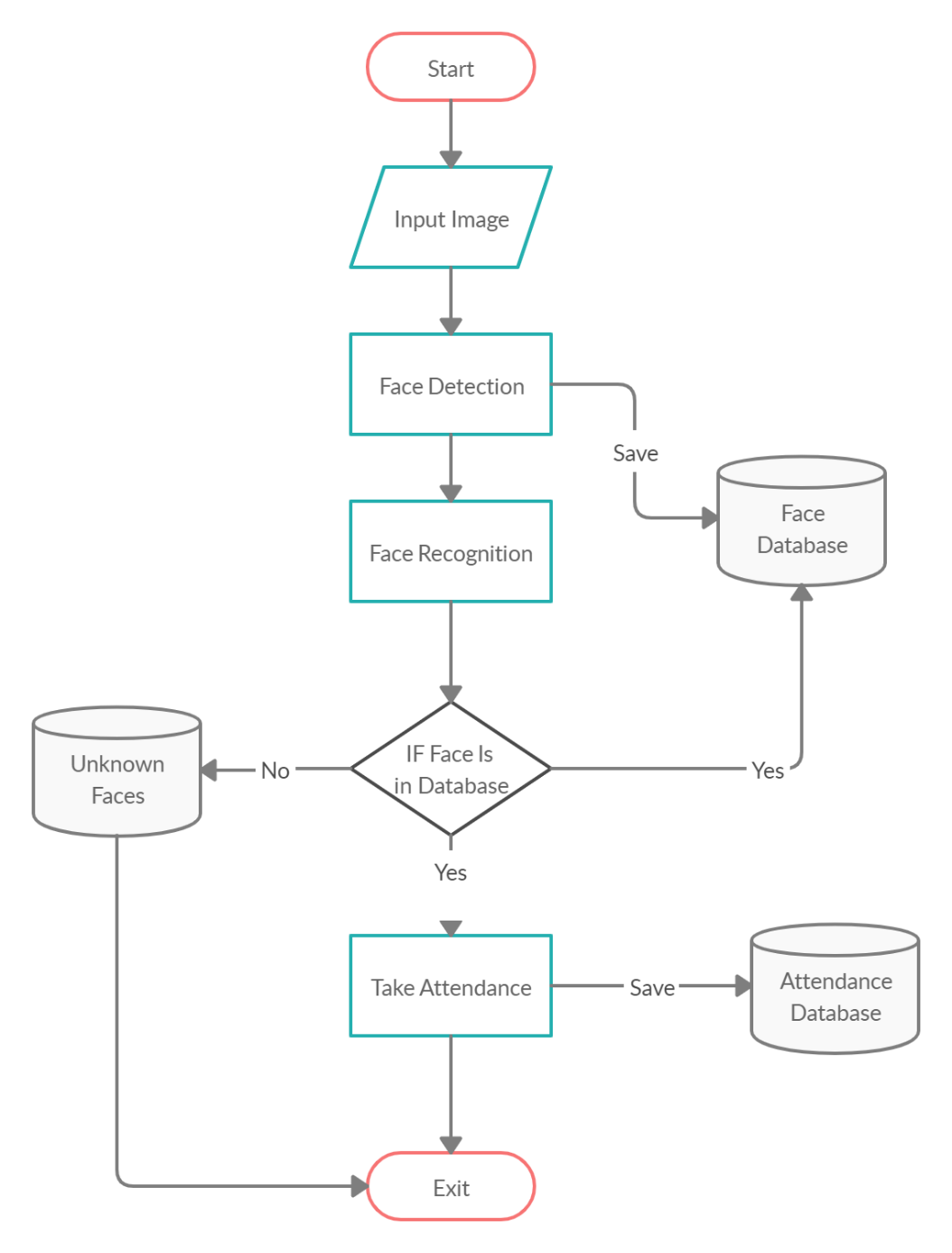
• Phát hiện phần khuôn mặt từ khung hình video.

• Trích xuất các tính năng hữu ích từ khuôn mặt được phát hiện.

• Phân loại các tính năng để nhận diện khuôn mặt đượcphát hiện.

• Ghi lại sự tham dự, có mặt của người được xác định.

* 1. **Lưu đồ thuật toán**



# **Chương 2: Xem xét tài liệu**

# **2.1 Xử lí ảnh kỹ thuật số**

Xử lý hình ảnh kỹ thuật số là việc sử dụng các thuật toán trên máy tính để thực hiện xử lý hình ảnh trên hình ảnh kỹ thuật số. Kỹ thuật xử lý ảnh kỹ thuật số được triển khai trong bốn ứng dụng chính là chủ yếu:

+, Thị giác máy tính

+, Truyền và mã hóa

+, Cải thiện và khôi phục hình ảnh

+, Lĩnh vực nhận dạng

## **2.2 Đại diện hình ảnh trong máy tính kỹ thuật số**

Một bức ảnh là một chức năng cường độ ánh sáng 2 chiều (2D) được ký hiệu:

𝐟 (𝐱,𝐲) = 𝐫 (𝐱,𝐲) × 𝐢 (𝐱,𝐲)

Với 𝐫 (𝐱,𝐲) là độ phản xạ bề mặt điểm ảnh tương ứng, 𝐢 (𝐱,𝐲) là cường độ ánh sáng tới. Một hình ảnh kỹ thuật số f (x, y) bị rời rạc cả về tọa độ không gian theo lưới và độ sáng bằng lượng tử hóa.

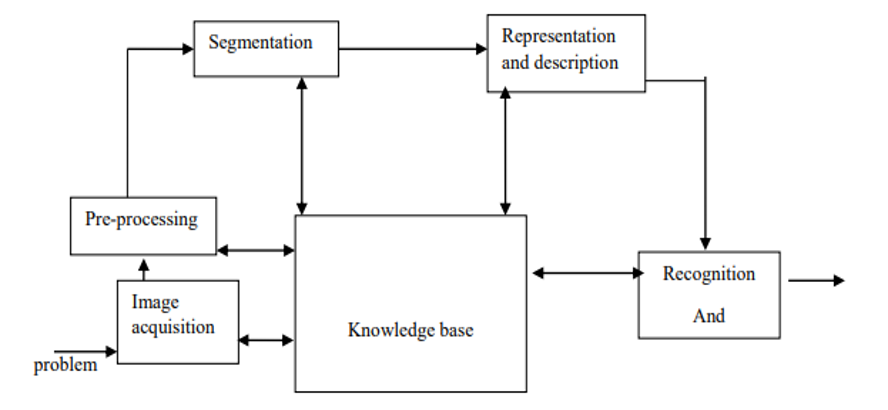
Thực tế, hình ảnh có thể được biểu diễn dưới dạng ma trận có hàng, chỉ số cột xác định một điểm trong ảnh và giá trị phần tử xác định giá trị mức xám tại điểm đó. Các yếu tố này được gọi là pixel hoặc pels.

Thông thường sau các ứng dụng xử lý hình ảnh, kích thước hình ảnh được sử dụng là 𝟐𝟓𝟔 × 𝟐𝟓𝟔, 𝟔𝟒𝟎 × 𝟒𝟖𝟎 pels hoặc 𝟏𝟎𝟐𝟒 × 𝟏𝟎𝟐𝟒 pixel. Lượng tử hóa các pixel ma trận này được thực hiện ở 8 bits cho hình ảnh đen trắng và 24s bit cho hình ảnh màu (vì ba mặt phẳng màu Đỏ, Xanh lục và Xanh lam mỗi màu 8 bits)

## **2.3 Các bước xử lí hình ảnh kỹ thuật số**

Xử lý ảnh kỹ thuật số bao gồm các tác vụ cơ bản sau:

* Thu nhận ảnh
* Tiền xử lí
* Phân đoạn
* Trích xuất mô tả của đối tượng hình ảnh
* Nhận diện và gán nhãn cho đối tượng dựa trên thông tin được cung cấp bởi mô tả của nó
* Cở sở tri thức - là sản phẩm cuối cùng của việc thu thập và sắp xếp tất cả các thông tin đó thành một hình thức hữu ích, thông qua một quy trình gọi là quản lý kiến thức



## **2.4 Định nghĩa**

### **2.4.1 Phát hiện khuôn mặt**

Nhận diện khuôn mặt là quá trình xác định và định vị tất cả khuôn mặt hiện tại trong một hình ảnh hoặc video bất kể vị trí, quy mô, hướng, tuổi và biểu cảm. Hơn nữa, việc phát hiện nên không liên quan đến chiếu sáng bên ngoài điều kiện và nội dung hình ảnh và video

### **2.4.2 Nhận diện khuôn mặt**

Nhận dạng khuôn mặt là một vấn đề nhận dạng mô hình hình ảnh, trong đó khuôn mặt, được biểu thị dưới dạng vật thể ba chiều có thể chiếu sáng khác nhau, tư thế và các yếu tố khác, cần được xác định dựa trên hình ảnh thu được.

Do đó, nhận dạng khuôn mặt chỉ đơn giản là nhiệm vụ xác định khuôn mặt đã được phát hiện là khuôn mặt đã biết hoặc chưa biết và trong các trường hợp nâng cao hơn cho biết chính xác khuôn mặt của ai.

### **2.4.3 Phân biệt giữa “Phát hiện khuôn mặt” và “Nhận diện khuôn mặt”**

“Phát hiện khuôn mặt” trả lời cho câu hỏi “Khuôn mặt ở đâu ?”. Nó xác định một đối tượng là khuôn mặt và định vị trong hình ảnh đầu vào.

“Nhận diện khuôn mặt” trả lời cho câu hỏi “Khuôn mặt của ai đây ?”. Nó quyết định xem khuôn mặt được phát hiện là ai đó đã biết hoặc chưa biết dựa trên cơ sở dữ liệu của khuôn mặt mà nó sử dụng để xác thực hình ảnh đầu vào. Do đó có thể thấy rằng đầu ra của “Phát hiện khuôn mặt” chính là đầu vào của “Nhận diện khuôn mặt”

### **2.4.4 Thuật toán phát hiện khuôn mặt LBPH ( Local Binary Patterns Histogram )**

Mô hình nhị phân cục bộ (LBP) là một toán tử kết cấu đơn giản nhưng rất hiệu quả, gắn nhãn các pixel của hình ảnh bằng cách chia vùng lân cận của mỗi pixel và coi kết quả là số nhị phân.

Nó được mô tả lần đầu tiên vào năm 1994 (LBP) và từ đó được phát hiện là một tính năng mạnh mẽ để phân loại kết cấu.

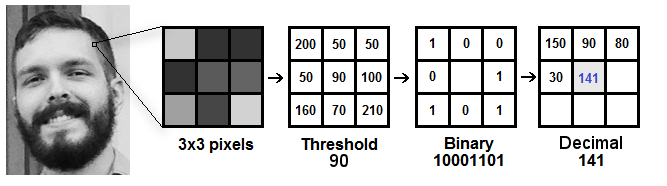
Người ta đã xác định thêm rằng khi LBP được kết hợp với biểu đồ mô tả độ dốc định hướng (HOG), nó sẽ cải thiện đáng kể hiệu suất phát hiện trên một số bộ dữ liệu. Sử dụng LBP kết hợp với histogram, chúng ta có thể biểu diễn hình ảnh khuôn mặt bằng một vectơ dữ liệu đơn giản.

Thuật toán LBPH hoạt động trong 5 bước:

1. **Parameter**: LBPH sử dụng 4 thông số

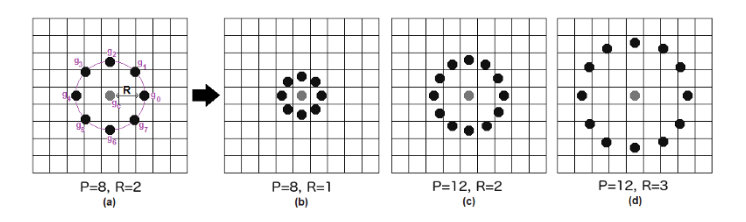
* **Bán kính**: bán kính được sử dụng để xây dựng LBP hình tròn và biểu thị bán kính xung quanh pixel trung tâm. Nó thường được đặt là 1.
* **Vùng lân cận**: số lượng điểm mẫu để xây dựng LBP hình tròn. Càng bao gồm nhiều điểm mẫu, chi phí tính toán càng cao. Nó thường được đặt đến 8.
* **Lưới X**: số lượng của ô theo chiều ngang. Càng nhiều ô, lưới càng mịn, thì chiều của kết quả vecto đặc trưng càng cao. Thường được đặt là 8.
* **Lưới Y**: số lượng của ô theo chiều dọc. Càng nhiều ô, lưới càng mịn, thì chiều của kết quả vecto đặc trưng càng cao. Thường được đặt là 8.

1. **Training**: Đầu tiên phải training thuật toán. Muốn vậy ta cần sử dụng một tập dữ liệu với hình ảnh khuôn mặt của những người chúng ta muốn nhận diện. Chúng ta cũng cần đặt ID ( có thể là số hoặc tên người ) cho mỗi hình ảnh, vì vậy thuật toán sẽ sử dụng thông tin này để nhận dạng hình ảnh đầu vào và cung cấp cho bạn đầu ra. Hình ảnh của cùng một người phải có cùng ID. Với bộ training đã được xây dựng sẵn, hãy xem các bước tính toán LBPH
2. **Phép tính LBP**: Bước tính toán đầu tiên của LBPH là để tạo ra hình ảnh trung gian mô ta hình ảnh gốc theo cách tốt hơn, bằng cách làm nổi bật đặc điểm khuôn mặt. Để làm vậy thuật toán sự dụng khái niệm về cửa sổ trượt, dựa trên tham số bán kính và vùng lân cận.



Dựa vào hình ảnh trên, hãy chia thành các bước nhỏ để có thể hiểu dễ dàng hơn:

* Giả sử ta có một bức ảnh khuôn mặt màu xám
* Chúng ta lấy 1 phần bức ảnh dưới dạng cửa sổ 3x3 pixels
* Nó cũng có thể được biểu diễn dưới dạng ma trận 3x3 chứa giá trị điểm ảnh của từng pixel (0 ~ 255).
* Sau đó, chúng ta cần lấy giá trị trung tâm của ma trận dùng làm ngưỡng.
* Giá trị này sẽ được sử dụng để xác định các giá trị mới từ 8 điểm vùng lân cận
* Đối với mỗi lân cận của giá trị trung tâm (ngưỡng), chúng ta đặt một giá trị nhị phân mới. Chúng ta đặt 1 cho các giá trị bằng hoặc cao hơn ngưỡng, và 0 cho các giá trị thấp hơn ngưỡng
* Bây giờ ma trận sẽ chỉ chứa các giá trị nhị phân ( bỏ qua giá trị ngưỡng ). Chúng ta cần nối từng giá trị nhị phân từ mỗi vị trí từ dòng ma trận theo từng dòng thành một giá trị nhị phân mới ( có thể nối theo hướng khác nhưng kết quả cuối cùng sẽ là tương tự )
* Sau đó chúng ta chuyển đổi giá trị nhị phân này thành giá trị thập phân và đặt nó thành giá trị trung tâm của ma trận.
* Vào cuối quy trình này (thủ tục LBP), chúng ta có một hình ảnh mới thể hiện rõ hơn các đặc điểm của hình ảnh gốc.



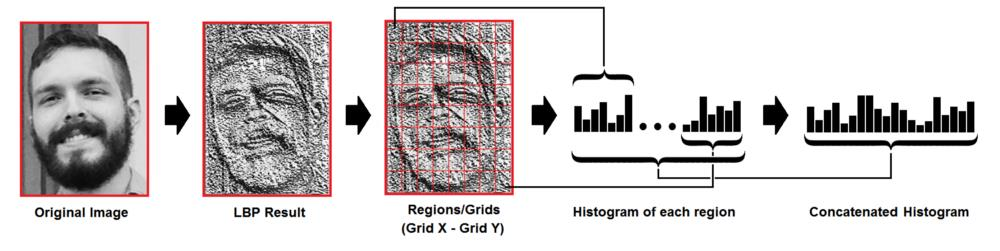
Quá trình thay đổi bán kính ( LBP )

Nó có thể được thực hiện bằng cách sử dụng phép nội suy song tuyến.

Nếu một số điểm dữ liệu nằm giữa các pixel,

nó sử dụng các giá trị từ 4 pixel gần nhất (2x2) để ước tính giá trị của điểm dữ liệu mới.

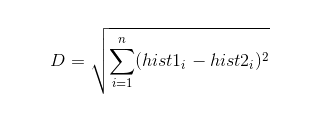
1. **Trích xuất Histogram**: Bây giờ, sử dụng hình ảnh được tạo ở bước cuối cùng, chúng ta có thể sử dụng các tham số Grid X và Grid Y để chia hình ảnh thành nhiều lưới, như có thể thấy trong hình ảnh sau đây:



Dựa vào hình ảnh trên, chúng ta có thể trích xuất Histogram của từng vùng như sau:

* Chúng ta có 1 ảnh xám, mỗi Histogram ( từ mỗi lưới ) sẽ chỉ chứa 256 vị trí ( 0 ~ 255 ) đại diện cho các lần xuất hiện của từng cường độ pixel.
* Sau đó chúng ta cần nối từng biểu đồ để tạo ra một biển đồ mới và lớn hơn. Giả sử chúng ta có lưới 8x8, chúng ta sẽ có 8x8x256 = 16384 vị trí trong histogram cuối cùng. Histogram cuối cùng đại diện có các đặc điểm của hình ảnh gốc.

1. **Thực hiện nhận diện khuôn mặt**: Trong bước này, thuật toán đã được đào tạo. Mỗi histogram được tạo ra được sử dụng để thể hiện mỗi hình ảnh từ tập dữ liệu đào tạo. Vì vậy, được cung cấp một hình ảnh đầu vào, chúng ta thực hiện lại các bước cho hình ảnh mới này và tạo ra một histogram đại diện cho hình ảnh.
   * Vì vậy, để tìm hình ảnh phù hợp với hình ảnh đầu vào, chúng ta chỉ cần so sánh 2 histogram với nhau và trả lại hình ảnh với các histogram gần giống nhau nhất.
   * Chúng ta có thể sử dụng nhiều cách để so sánh histogram ( tính khoảng cách giữa 2 histogram ), ví dụ: Khoảng cách euclide, bình phương, giá trị tuyệt đối,… Trong ví dụ vày, chúng ta có thể sử dụng khoảng cách euclide dựa trên công thức sau:



* Và đầu ra của thuật toán này sẽ là ID từ bức ảnh với histogram gần giống nhất. Thuật toán cũng sẽ trả về khoảng cách được tính toán, có thể được sử dụng như một phép đo độ tin cậy.
* Sau đó, chúng tôi có thể sử dụng ngưỡng và ‘độ tin cậy để tự động ước tính nếu thuật toán đã nhận dạng chính xác hình ảnh. Chúng ta có thể giả định rằng thuật toán đã được công nhận thành công nếu độ tin cậy thấp hơn ngưỡng được xác định.

# **Chương 3: Mô hình thực hiện và phân tích**

## **3.1 Mô hình thực hiện**

Công cụ chính được sử dụng trong phương pháp triển khai là thư việc thị giác máy tính nguồn mở OpenCV. OpenCV (Open Computer Vision) là một thư viện mã nguồn mở chuyên dùng để xử lý các vấn đề liên quan đến thị giác máy tính. Nhờ một hệ thống các giải thuật chuyên biệt, tối ưu cho việc xử lý thị giác máy tính, vì vậy tính ứng dụng của OpenCV là rất lớn. Thư viện OpenCV chứa hơn 500 hàm trải rộng trên nhiều lĩnh vực trong thị giác. Công nghệ chính đằng sau nhận diện khuôn mặt là OpenCV. Người đứng trước camera phải giữ khoảng cách tối thiểu là 50cm và hình ảnh được chụp sẽ là đầu vào. Mặt trước được trích xuất thành hình ảnh sau đó chuyển sang màu xám và được lưu trữ. Thuật toán “Phân tích thành phần chính” ( PCA ) được thực hiện trên ảnh và các giá trị riêng được lưu trong 1 file xml. Khi người dùng yêu cầu nhận dạng, mặt trước được trích xuất từ khung hình video đã chụp qua camera. Giá trị riêng được tính lại cho mặt kiểm tra và nó được khớp với dữ liệu được lưu trữ cho vùng lân cận gần giống nhất.

### **3.1.1 Phần mềm triển khai**

* OpenCV
* Visual Studio Code
* Python IDE
* Anaconda 3

### **3.1.2 Phần cứng triểu khai**

* Laptop acer A515
  + CPU intel core i5, 7200U, 2.50 GHz
  + RAM 4GB
  + Webcam 1MP, HD Webcam

## **3.2 Kết quả thực nghiệm**

**Phát hiện khuôn mặt**

Bắt đầu chụp ảnh từ camera

Bắt đầu:

* Xử lí trước hình ảnh đã chụp và trích xuất hình ảnh khuôn mặt
* Tính giá trị riêng của hình ảnh khuôn mặt được chụp và so sánh với giá trị riêng của các khuôn mặt hiện có trong cơ sở dữ liệu
* Nếu giá trị riêng không khớp với giá trị hiện tại thì sẽ lưu thông tin hình ảnh khuôn mặt mới vào cơ sở dữ liệu ( tệp xml )
* Nếu giá trị riêng khớp với giá trị riêng hiện có thì bước nhận diện sẽ được thực hiện

Kết thúc.

**Nhận diện khuôn mặt**

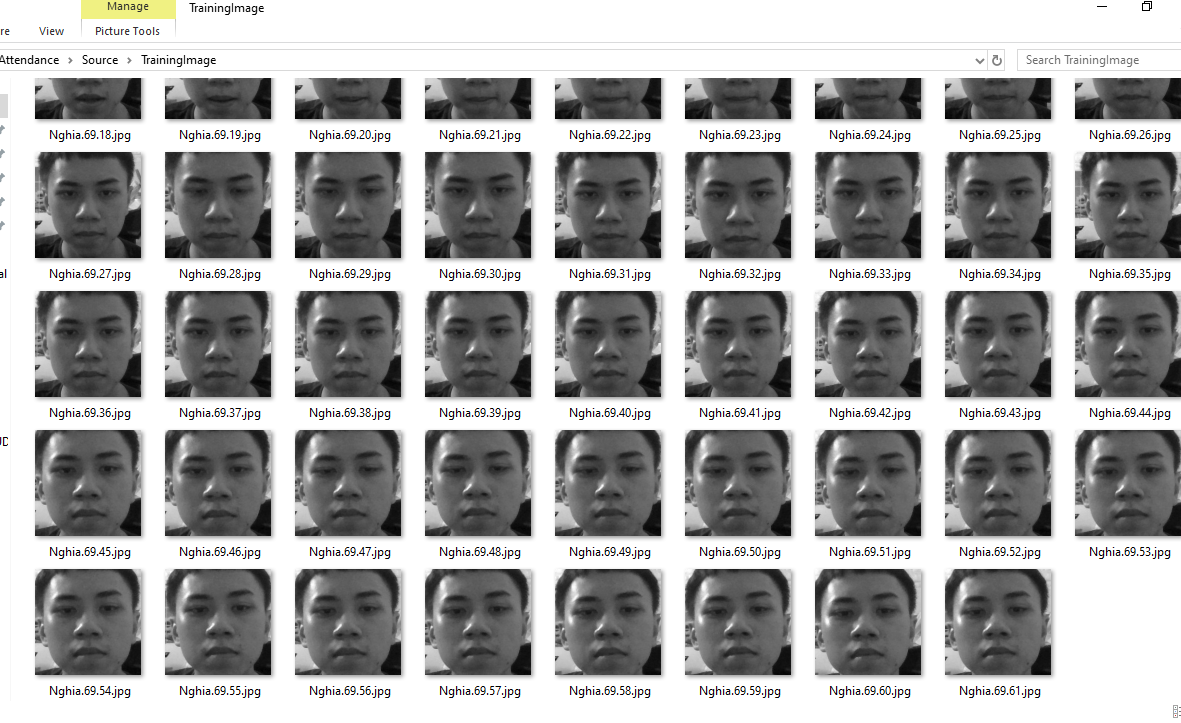
Sử dụng thuật toán PCA, các bước sau sẽ được thực hiện cho bước nhận diện khuôn mặt:

Bắt đầu:

* Tìm thông tin của khuôn mặt từ ảnh phù hợp với cơ sở dữ liệu
* Cập nhật bảng ghi chú với hình ảnh khuôn mặt tương ứng và thời gian hệ thống hoàn thành việc điểm danh cho người chụp hình được.

Kết thúc.

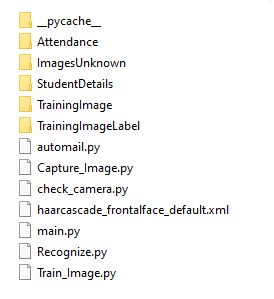
Kết quả chụp ảnh khuôn mặt thành ảnh xám tỉ lệ 50x50 pixel



# **Chương 4: Tiến hành lập trình**

## **4.1 Tiến hành lập trình**

Tất cả code được viết bằng ngôn ngữ Python. Dưới đây là cấu trúc thư mục:



Sau khi tạo môi trường cần thiết để tiến hành lập trình, hãy cài đặt các gói cần thiết. Để cài đặt, mở Anaconda Prompt và gõ lệnh:

pip install opencv-contrib-python

pip install numpy

pip install pandas

pip install Pillow

pip install pytest-shutil

pip install python-csv

pip install yagmail

### **4.1.1 haarcascade\_frontalface\_defaultx.xml**

Đây là file hỗ trợ, cho phép xây dựng hàm phát hiện khuôn mặt. Diễn giải theo cách đơn giản, nó áp dụng bộ lọc trên cửa sổ trượt của 1 khung hình. Trong mỗi lần trượt qua 1 vùng khung hình, từng bước một sẽ được xử lí: nếu bước 1 nhận đó là mặt, chúng ta chuyển qua bước 2; và nếu không thì chúng ta bỏ qua vùng đó và trượt cửa sổ đi chỗ khác. Nếu một vùng pass toàn bộ các bước test mặt đó thì cửa sổ đó có chứa mặt người. Nếu có các ô đè lên nhau cùng chứa một mặt, các toạ độ các góc sẽ được cộng vào lấy trung bình.

### **4.1.2 main.py**

Tất cả công việc được thực hiện ở đây, phát hiện khuôn mặt, nhận diện khuôn mặt, hoàn thành điểm danh.

import os

import check\_camera

import Capture\_Image

import Train\_Image

import Recognize

def title\_bar():

os.system('cls')

print("\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("\t\*\*\*\*\* Attendance System \*\*\*\*\*")

print("\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

def mainMenu():

title\_bar()

print()

print(10 \* "\*", "WELCOME MENU", 10 \* "\*")

print("[1] Check Camera")

print("[2] Capture Faces")

print("[3] Train Images")

print("[4] Recognize & Attendance")

print("[5] Auto Mail")

print("[6] Quit")

while True:

try:

choice = int(input("Enter Choice: "))

if choice == 1:

checkCamera()

break

elif choice == 2:

CaptureFaces()

break

elif choice == 3:

Trainimages()

break

elif choice == 4:

RecognizeFaces()

break

elif choice == 5:

os.system("python automail.py")

break

mainMenu()

elif choice == 6:

print("Thank You")

break

else:

print("Invalid Choice. Enter 1-4")

mainMenu()

except ValueError:

print("Invalid Choice. Enter 1-4\n Try Again")

exit

def checkCamera():

check\_camera.camer()

key = input("Enter any key to return main menu")

mainMenu()

def CaptureFaces():

Capture\_Image.takeImages()

key = input("Enter any key to return main menu")

mainMenu()

def Trainimages():

Train\_Image.TrainImages()

key = input("Enter any key to return main menu")

mainMenu()

def RecognizeFaces():

Recognize.recognize\_attendence()

key = input("Enter any key to return main menu")

mainMenu()

mainMenu()

### 

### **4.1.3 Check\_camera.py**

Chức năng kiểm tra camera webcam.

import cv2

def camer():

cap = cv2.VideoCapture(0)

while(True):

ret, frame = cap.read()

gray = cv2.cvtColor(frame,cv2.COLOR\_BGR2GRAY)

cv2.imshow('frame', gray)

if cv2.waitKey(1) & 0xFF == ord('q'):

break

cap.release()

cv2.destroyAllWindows()

### **4.1.4 Capture\_Image.py**

Nhiệm vụ chụp ảnh khuôn mặt có trong khung video camera và thông tin ID, name của khuôn mặt đó tất cả được lưu lại để chuẩn bị cho việc training image.

import csv

import cv2

import os

def is\_number(s):

try:

float(s)

return True

except ValueError:

pass

try:

import unicodedata

unicodedata.numeric(s)

return True

except (TypeError, ValueError):

pass

return False

def takeImages():

Id = input("Enter Your Id: ")

name = input("Enter Your Name: ")

if(is\_number(Id) and name.isalpha()):

cam = cv2.VideoCapture(0)

harcascadePath = "haarcascade\_frontalface\_default.xml"

detector = cv2.CascadeClassifier(harcascadePath)

sampleNum = 0

while(True):

ret, img = cam.read()

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

faces = detector.detectMultiScale(gray, 1.3, 5)

for(x,y,w,h) in faces:

cv2.rectangle(img, (x, y), (x+w, y+h), (255, 0, 0), 2)

sampleNum = sampleNum+1

cv2.imwrite("TrainingImage" + os.sep +name + "."+Id + '.' +

str(sampleNum) + ".jpg", gray[y:y+h, x:x+w])

cv2.imshow('frame', img)

if cv2.waitKey(100) & 0xFF == ord('q'):

break

elif sampleNum > 60:

break

cam.release()

cv2.destroyAllWindows()

res = "Images Saved for ID : " + Id + " Name : " + name

row = [Id, name]

with open("StudentDetails"+os.sep+"StudentDetails.csv", 'a+') as csvFile:

writer = csv.writer(csvFile)

writer.writerow(row)

csvFile.close()

else:

if(is\_number(Id)):

print("Enter Alphabetical Name")

if(name.isalpha()):

print("Enter Numeric ID")

### **4.1.5 Train\_Image.py**

Chứa 2 hàm chính là hàm train image, gán label cho các ảnh với dữ liệu từ folder “TrainingImage”. Kết quả sẽ được lưu vào file “Trainer.yml” nằm trong thư mục “TrainingImageLabel”.

import os

import cv2

import numpy as np

from PIL import Image

def getImagesAndLabels(path):

imagePaths = [os.path.join(path, f) for f in os.listdir(path)]

faces = []

Ids = []

for imagePath in imagePaths:

pilImage = Image.open(imagePath).convert('L')

imageNp = np.array(pilImage, 'uint8')

Id = int(os.path.split(imagePath)[-1].split(".")[1])

faces.append(imageNp)

Ids.append(Id)

return faces, Ids

def TrainImages():

recognizer = cv2.face\_LBPHFaceRecognizer.create()

harcascadePath = "haarcascade\_frontalface\_default.xml"

detector = cv2.CascadeClassifier(harcascadePath)

faces, Id = getImagesAndLabels("TrainingImage")

recognizer.train(faces, np.array(Id))

recognizer.save("TrainingImageLabel"+os.sep+"Trainner.yml")

print("Images Trained")

### **4.1.6 Recognize.py**

Chứa hàm nhận diện và điểm danh.

import datetime

import os

import time

import cv2

import pandas as pd

def recognize\_attendence():

recognizer = cv2.face.LBPHFaceRecognizer\_create()

recognizer.read("TrainingImageLabel"+os.sep+"Trainner.yml")

harcascadePath = "haarcascade\_frontalface\_default.xml"

faceCascade = cv2.CascadeClassifier(harcascadePath)

df = pd.read\_csv("StudentDetails"+os.sep+"StudentDetails.csv")

cam = cv2.VideoCapture(0)

font = cv2.FONT\_HERSHEY\_SIMPLEX

col\_names = ['Id', 'Name', 'Date', 'Time']

attendance = pd.DataFrame(columns=col\_names)

while True:

ret, im = cam.read()

gray = cv2.cvtColor(im, cv2.COLOR\_BGR2GRAY)

faces = faceCascade.detectMultiScale(gray, 1.2, 5)

for(x, y, w, h) in faces:

cv2.rectangle(im, (x, y), (x+w, y+h), (225, 0, 0), 2)

Id, conf = recognizer.predict(gray[y:y+h, x:x+w])

if(conf < 50):

ts = time.time()

date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')

timeStamp = datetime.datetime.fromtimestamp(

ts).strftime('%H:%M:%S')

aa = df.loc[df['Id'] == Id]['Name'].values

tt = str(Id)+"-"+aa

attendance.loc[len(attendance)] = [Id, aa, date, timeStamp]

else:

Id = 'Unknown'

tt = str(Id)

if(conf > 75):

noOfFile = len(os.listdir("ImagesUnknown"))+1

cv2.imwrite("ImagesUnknown"+os.sep+"Image"+str(noOfFile) +

".jpg", im[y:y+h, x:x+w])

cv2.putText(im, str(tt), (x, y+h), font, 1, (255, 255, 255), 2)

attendance = attendance.drop\_duplicates(subset=['Id'], keep='first')

cv2.imshow('im', im)

if (cv2.waitKey(1) == ord('q')):

break

ts = time.time()

date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')

timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')

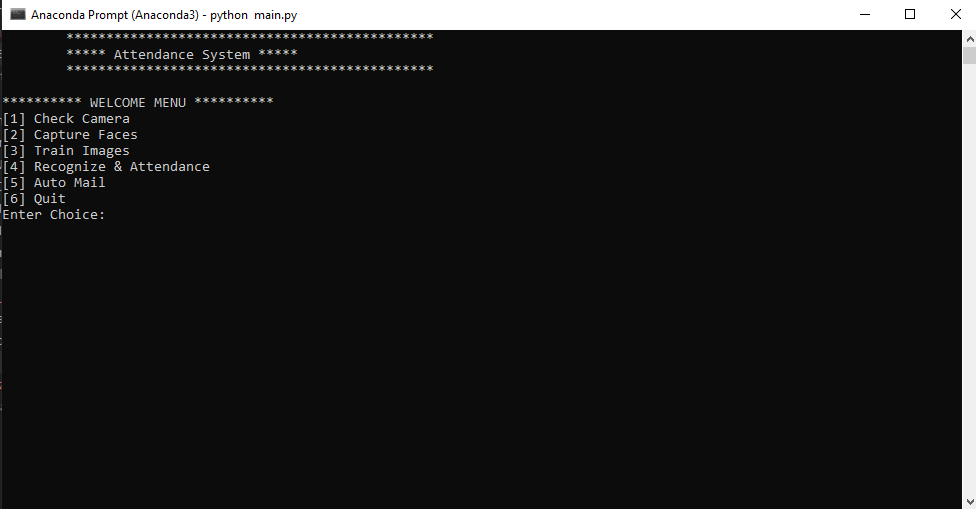
Hour, Minute, Second = timeStamp.split(":")

fileName = "Attendance"+os.sep+"Attendance\_"+date+"\_"+Hour+"-"+Minute+"-"+Second+".csv"

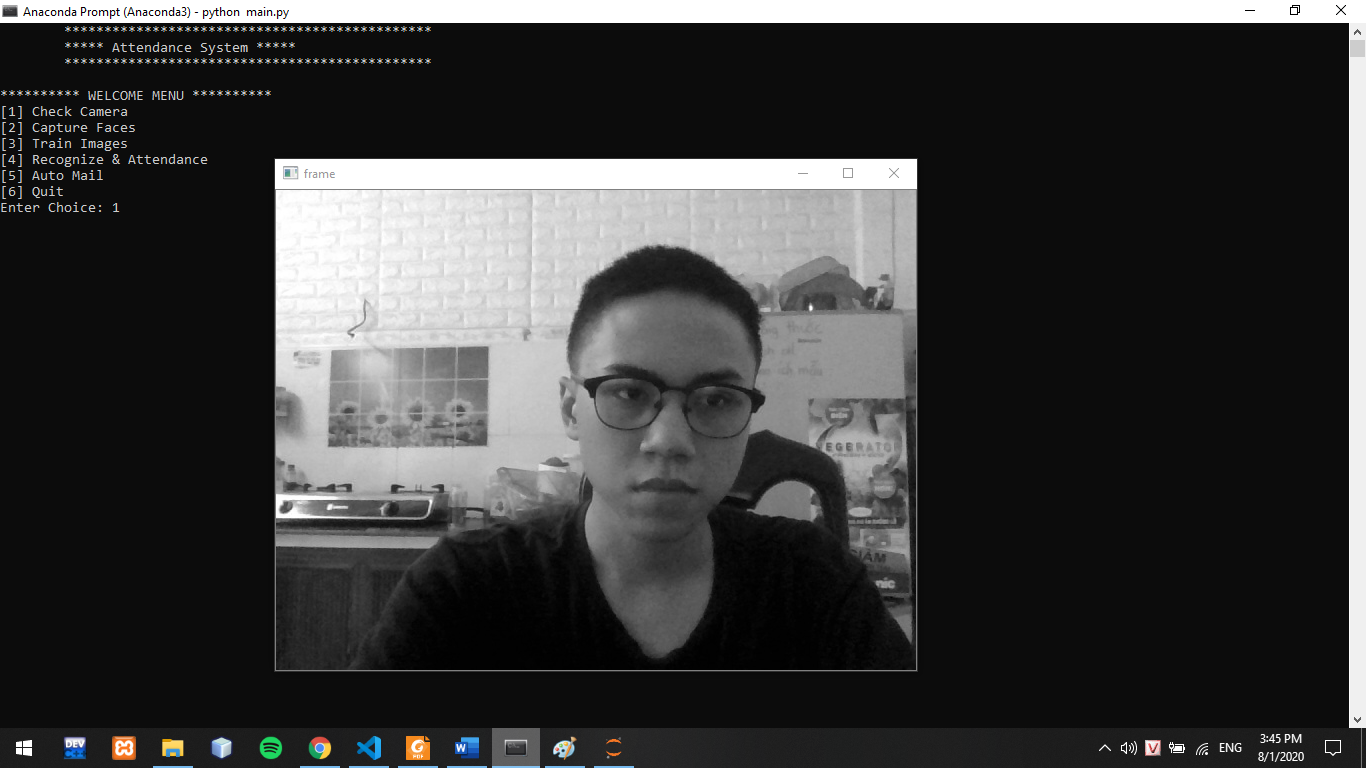
attendance.to\_csv(fileName, index=False)

# **Chương 5: Tổng kết**

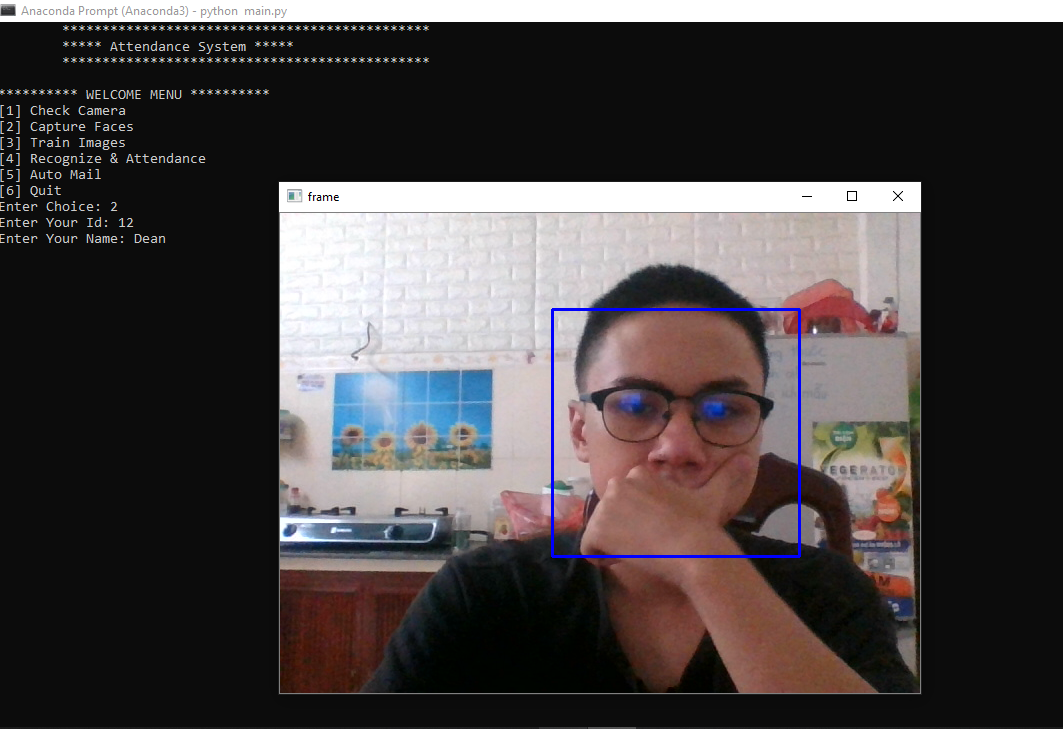
## **5.1 Hình ảnh kết quả chương trình**

****

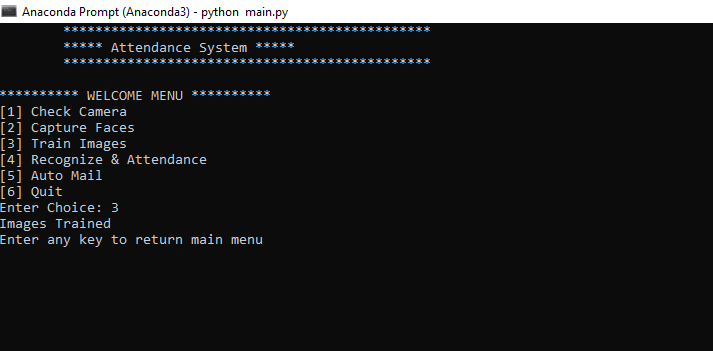
***Giao diện chương trình***



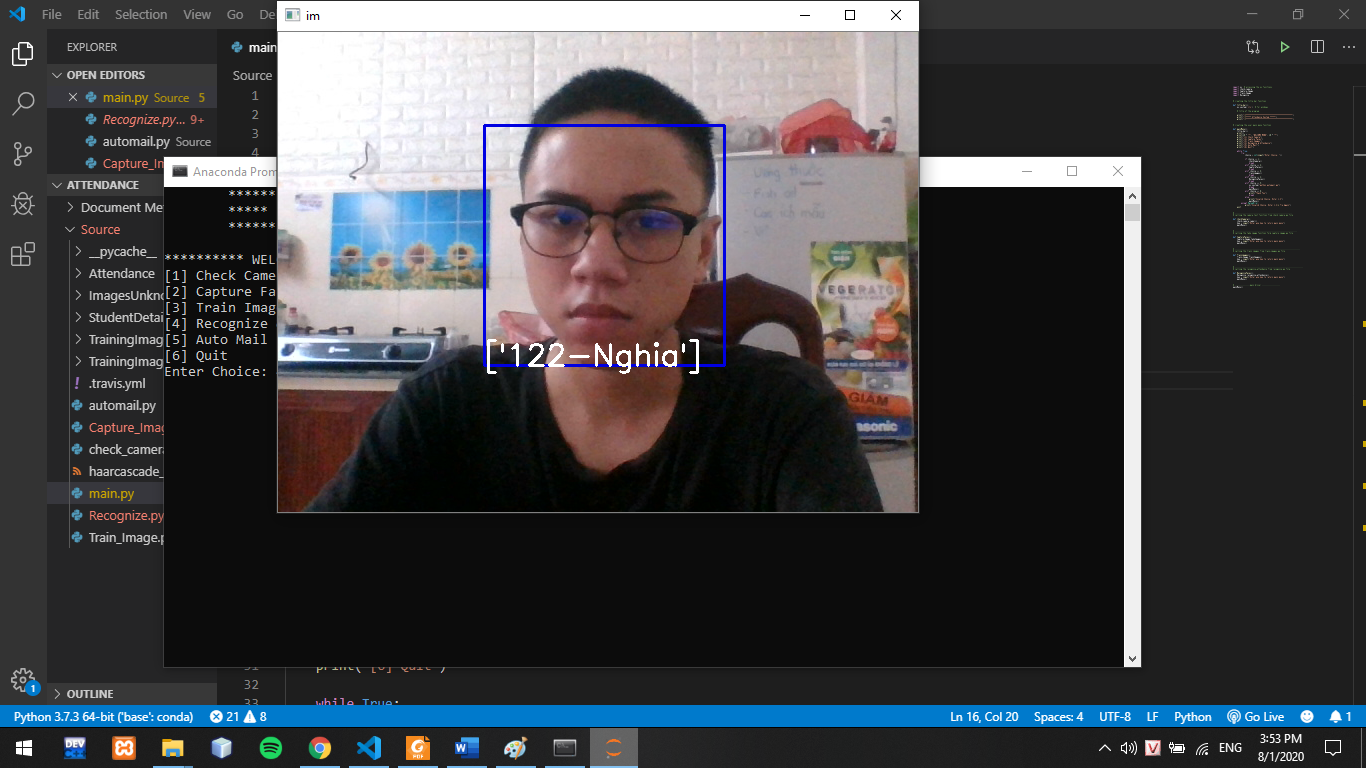
***Check Camera***



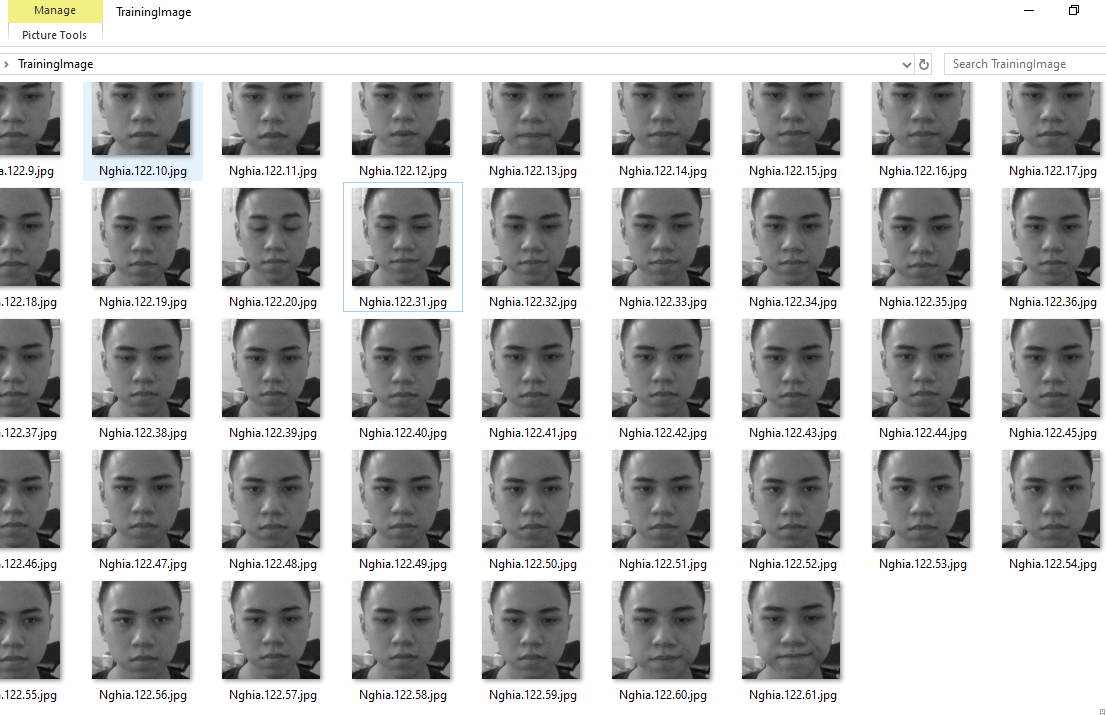
***Capture Faces***



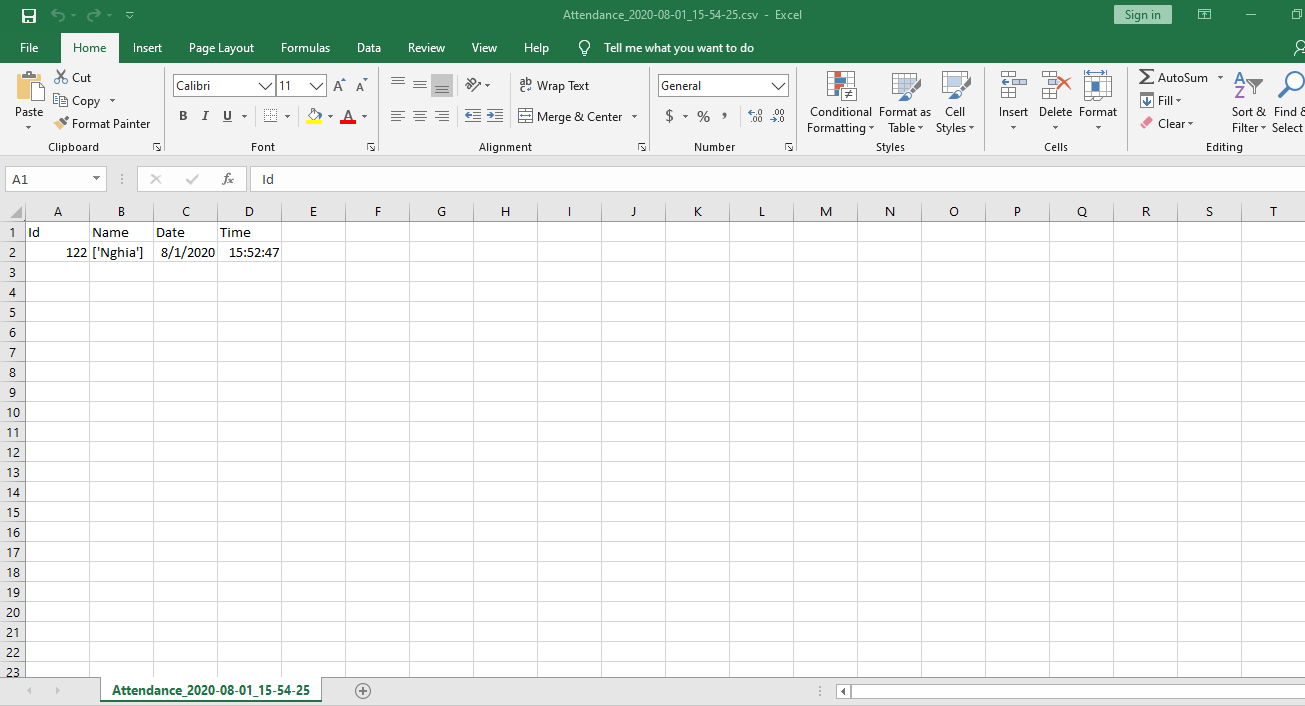
***Train Image***



***Recognize & Attendance***



***Training Image***



***Attendance file excel***

## **5.2 Ưu điểm và hạn chế của dự án**

### **5.2.1 Ưu điểm**

Chương trình đã có thể phát hiện khuôn mặt, nhận nhiện khuôn mặt đó là ai, thực hiện việc điểm danh, chấm công ở mức khá hoàn thiện. Về giao diện, đây là 1 chương trình có giao diện đơn giản, dễ sử dụng đối với tất cả mọi người, kể cả một người không rành quá nhiều về công nghệ thông tin. Và cơ bản chương trình đã có thể được sử dụng tại một số môi trường, phạm vi nhất định

### **5.2.2 Hạn chế**

Điểm hạn chế của dự án này là chúng tôi vẫn chưa tiến hành trên các thiết bị khác. Tính năng gửi email tệp điểm danh tự động chưa hoàn thiện và tính bảo mật của chương trình vẫn chưa có

## **5.3 Hướng phát triển**

Trong tương lai, chúng tôi sẽ khắc phục các hạn chế còn tồn tại của dự án đồng thời nghiên cứu, phát triển thêm các tính năng mới. Cụ thể:

* Hoàn thiện tính năng gửi email tệp điểm danh tự động
* Cải thiện tính an ninh và bảo mật cho chương trình
* Có thể sử dụng mạng Nơ ron để cho ra kết quả chính xác cao hơn
* Có thể xây dựng trên hệ thống website

# 

# **Tài liệu tham khảo:**

<https://www.pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/>

<https://github.com/kmbin/Face-Recognition-System>