

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



**MÔN: XỬ LÝ ẢNH**

**ĐỀ TÀI: STUDENT ATTENDANCE CHECKING**

**(web-based)**

|  |  |
| --- | --- |
| **Giảng viên**  **Nhóm lớp**  **Sinh viên**  **Nguyễn Công Lâm**  **Bùi Minh Quý**  **Tô Minh Hiếu** | **: Phạm Văn Sự**  **:09**  **Mã sinh viên**  **B20DCCN391**  **B20DCDT171**  **B20DCDT076** |

**Hà Nội, tháng 12, năm 2023**

# BẢNG PHÂN CÔNG CÔNG VIỆC

|  |  |  |  |
| --- | --- | --- | --- |
| **Họ và tên** | **Nhiệm vụ** | **Đánh giá** | **Chấm điểm** |
| Nguyễn Công Lâm  (Nhóm trưởng) | - Chia phần việc cho các thành viên  - Triển khai bài toán  - Code chính  - Demo | Hoàn thành công việc. | 9 |
| Tô Minh Hiếu | - Tìm hiểu lý thuyết nhận dạng vật thể  -Soạn báo cáo | Hoàn thành công việc. | 8 |
| Bùi Minh Quý | - Tìm hiểu cách xây dựng giao diện  - Soạn báo cáo | Hoàn thành công việc. | 8 |

***Mã đề :*** DIPFL23PRJG47

***Đề bài :*** STC21: Student attendance checking (web-based)

**MỤC LỤC**

[BẢNG PHÂN CÔNG CÔNG VIỆC 2](#_Toc152789401)

[**LỜI MỞ ĐẦU** 4](#_Toc152789402)

[**CHƯƠNG 1 : TỔNG QUAN VỀ XỬ LÝ ẢNH VÀ NHẬN DẠNG VẬT THỂ** 5](#_Toc152789403)

[**1.1** **Tổng quan xử lý ảnh** 5](#_Toc152789404)

[**1.2** **Các vấn đề cơ bản trong xử lý ảnh** 5](#_Toc152789405)

[1.2.1 Một số khái niệm cơ bản 6](#_Toc152789406)

[1.2.2 Nắn chỉnh biến dạng 6](#_Toc152789407)

[1.2.3 Khử nhiễu 6](#_Toc152789408)

[1.2.4 Chỉnh mức xám 6](#_Toc152789409)

[1.2.5 Trích chọn đặc điểm 7](#_Toc152789410)

[1.2.6 Nhận dạng 7](#_Toc152789411)

[1.2.7 Nén ảnh 9](#_Toc152789412)

[**1.3 Nhận dạng vật thể** 9](#_Toc152789413)

[**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT NHẬN DẠNG VẬT THỂ** 10](#_Toc152789414)

[**2.1 Phát hiện khuôn mặt (face recognition) và nhận diện khuôn mặt (face verification)** 10](#_Toc152789415)

[2.1.1 Face Recogniton 11](#_Toc152789416)

[2.1.2 Face verification 12](#_Toc152789417)

[**2.2 Facenet và MTCNN** 12](#_Toc152789418)

[2.2.1 MTCNN 12](#_Toc152789419)

[2.2.2 FaceNet 14](#_Toc152789421)

[**CHƯƠNG 3: XÂY DỰNG GIAO DIỆN NHẬN DẠNG VẬT THỂ** 16](#_Toc152789422)

[**3.1 Công nghê sử dụng** 16](#_Toc152789423)

[3.1.1 HTML 16](#_Toc152789424)

[3.1.1.1 Khái niệm 16](#_Toc152789425)

[3.1.2 CSS 16](#_Toc152789426)

[3.1.3 JavaScript 17](#_Toc152789427)

[3.1.4 API 18](#_Toc152789428)

[3.1.5 SpringBoot 19](#_Toc152789429)

[3.1.6 Thư viện openCV 20](#_Toc152789430)

[3.1.7 Thư viện Numpy 21](#_Toc152789431)

[3.1.8 Các tiện ích khác 21](#_Toc152789432)

[**3.2 Kiến trúc hiện thống** 22](#_Toc152789433)

[**3.3 Quá trình chạy của hệ thống** 22](#_Toc152789434)

[**3.4 Kết quả khi chạy chương trình** 27](#_Toc152789435)

# **LỜI MỞ ĐẦU**

Trong những năm gần đây, xử lý hình ảnh (image processing) đã đạt được nhiều thành tựu và tiến bộ vượt bậc. Trong đó, nhận dạng và phân loại hình ảnh là một trong những lĩnh vực được theo đuổi một cách tích cực nhất. Ý tưởng cốt lõi của việc nhận dạng và phân loại hình ảnh là phân tích ảnh từ các dữ liệu thu được bởi các cảm biến hình ảnh như camera, Webcam,… Nhờ các hệ thống xử lý ảnh, con người đã giảm được khối lượng công việc cũng như tăng sự chính xác trong việc đưa ra các quyết định liên quan đến xử lý hình ảnh trên nhiều lĩnh vực: quân sự và quốc phòng, các hệ thống kỹ nghệ hoá sinh, giải phẫu, các hệ thống giao thông thông minh, robotics, các hệ thống an ninh…

Nhận thấy đây là vấn đề rất thiết thực, với những kiến thức đã được trang bị trong quá trình học tập, chúng em chọn đề tài : “Xử lý ánh nhận diện khuôn mặt”. Về phạm vi thực hiện, do giới hạn phạm vi của môn học, chúng em thực xây dựng đơn giản dựa trên kiến thức đã học thị giác máy tính và xử lý ảnh, chúng em chỉ thực hiện nhận diện với số lượng mẫu hạn chế . Về khả năng mở rộng, đề tài có tính thực tiễn và khả năng mở rộng cao, khả năng ứng dụng rộng rãi .

Với bài báo cáo này, chúng em chia làm 3 chương :

+ Chương 1: Tổng quan về Xử lý ảnh, nhận diện vật thể

+ Chương 2: Cơ sở lý thuyết về nhận dạng vật thể

+ Chương 3: Xây dựng giao diện nhận dạng vật thể

# **CHƯƠNG 1 : TỔNG QUAN VỀ XỬ LÝ ẢNH VÀ NHẬN DẠNG VẬT THỂ**

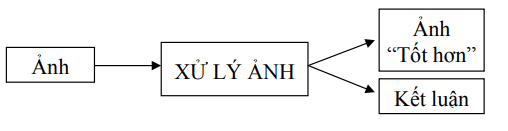
* 1. **Tổng quan xử lý ảnh**

Ảnh có thể xem là tập hợp các điểm ảnh và mỗi điểm ảnh được xem như là đặc trưng cường độ sáng hay một dấu hiệu nào đó tại một vị trí nào đó của đối tượng trong không gian và nó có thể xem như một hàm n biến P(c1, c2,..., cn). Do đó, ảnh trong xử lý ảnh có thể xem như ảnh n chiều

Ảnh số được định nghĩa là một hàm hai biến, F(s,t): I s, t: Các tọa độ ảnh, liên tục. I Giá trị hàm F: Cường độ sáng/mức xám/thông tin màu(độ lớn) tại điểm ảnh quan tâm, liên tục.

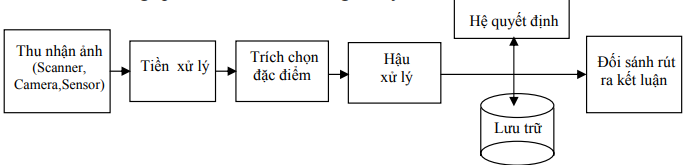
Xử lý ảnh số là quá trình xử lý ảnh số bằng công cụ máy tính điện tử, được xem như là quá trình thao tác ảnh đầu vào nhằm cho ra kết quả mong muốn. Kết quả đầu ra của một quá trình xử lý ảnh có thể là một ảnh “tốt hơn” hoặc một kết luận.

* Đầu vào: Dữ liệu ảnh số
* Đầu ra: Kết quả mong muốn Cải thiện thông tin ảnh, Thuận tiện lưu trữ/truyền tải, ...



Hình 1.1. Quá trình xử lý ảnh

Sơ đồ tổng quát của một hệ thống xử lý ảnh



Hình 1.2. Các bước cơ bản trong một hệ thống xử lý ảnh

* 1. **Các vấn đề cơ bản trong xử lý ảnh**

### *Một số khái niệm cơ bản*

\* Ảnh và điểm ảnh:

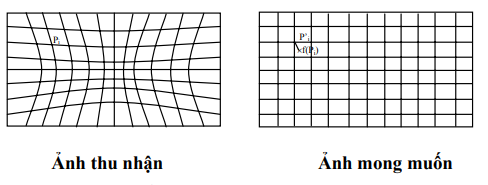
Điểm ảnh được xem như là dấu hiệu hay cường độ sáng tại 1 toạ độ trong không gian của đối tượng và ảnh được xem như là 1 tập hợp các điểm ảnh.

\* Mức xám, màu

Là số các giá trị có thể có của các điểm ảnh của ảnh

### *1.2.2 Nắn chỉnh biến dạng* Ảnh thu nhận thường bị biến dạng do các thiết bị quang học và điện

tử.



Hình 1.3. Ảnh thu nhận và ảnh mong muốn

### *1.2.3 Khử nhiễu*

Có 2 loại nhiễu cơ bản trong quá trình thu nhận ảnh

• Nhiều hệ thống: là nhiễu có quy luật có thể khử bằng các phép

biến đổi

• Nhiễu ngẫu nhiên: vết bẩn không rõ nguyên nhân → khắc phục

bằng các phép lọc

### *1.2.4 Chỉnh mức xám*

Nhằm khắc phục tính không đồng đều của hệ thống gây ra. Thông

thường có 2 hướng tiếp cận:

• Giảm số mức xám: Thực hiện bằng cách nhóm các mức xám gần

nhau thành một bó. Trường hợp chỉ có 2 mức xám thì chính là

chuyển về ảnh đen trắng. Ứng dụng: In ảnh màu ra máy in

đen trắng.

• Tăng số mức xám: Thực hiện nội suy ra các mức xám trung gian

bằng kỹ thuật nội suy. Kỹ thuật này nhằm tăng cường độ mịn

cho ảnh

### *1.2.5 Trích chọn đặc điểm*

Các đặc điểm của đối tượng được trích chọn tuỳ theo mục đích nhận

dạng trong quá trình xử lý ảnh. Có thể nêu ra một số đặc điểm của ảnh

sau đây:

**Đặc điểm không gian**: Phân bố mức xám, phân bố xác suất, biên độ,

điểm uốn v.v..

**Đặc điểm biến đổi**: Các đặc điểm loại này được trích chọn bằng việc

thực hiện lọc vùng (zonal filtering). Các bộ vùng được gọi là “mặt nạ đặc điểm” (feature mask) thường là các khe hẹp với hình dạng khác nhau (chữ nhật, tam giác, cung tròn v.v..)

**Đặc điểm biên và đường biên:** Đặc trưng cho đường biên của đối

tượng và do vậy rất hữu ích trong việc trích trọn các thuộc tính bất biến

được dùng khi nhận dạng đối tượng. Các đặc điểm này có thể được trích

chọn nhờ toán tử gradient, toán tử la bàn, toán tử Laplace, toán tử “chéo

không” (zero crossing) v.v..

Việc trích chọn hiệu quả các đặc điểm giúp cho việc nhận dạng các đối tượng ảnh chính xác, với tốc độ tính toán cao và dung lượng nhớ lưu trữ giảm xuống.

### *1.2.6 Nhận dạng*

Nhận dạng tự động (automatic recognition), mô tả đối tượng, phân

loại và phân nhóm các mẫu là những vấn đề quan trọng trong thị giác máy,

được ứng dụng trong nhiều ngành khoa học khác nhau. Tuy nhiên, một câu

hỏi đặt ra là: mẫu (pattern) là gì? Watanabe, một trong những người đi đầu

trong lĩnh vực này đã định nghĩa: “Ngược lại với hỗn loạn (chaos), mẫu là

một thực thể (entity), được xác định một cách ang áng (vaguely defined) và

có thể gán cho nó một tên gọi nào đó”. Ví dụ mẫu có thể là ảnh của vân tay,

ảnh của một vật nào đó được chụp, một chữ viết, khuôn mặt người hoặc

một ký đồ tín hiệu tiếng nói. Khi biết một mẫu nào đó, để nhận dạng hoặc phân loại mẫu đó có thể:

Hoặc **phân loại có mẫu** (supervised classification), chẳng hạn phân

tích phân biệt (discriminant analyis), trong đó mẫu đầu vào được định danh

như một thành phần của một lớp đã xác định.

Hoặc **phân loại không có mẫu** (unsupervised classification hay

clustering) trong đó các mẫu được gán vào các lớp khác nhau dựa trên một

tiêu chuẩn đồng dạng nào đó. Các lớp này cho đến thời điểm phân loại vẫn

chưa biết hay chưa được định danh.

Hệ thống nhận dạng tự động bao gồm ba khâu tương ứng với ba giai

đoạn chủ yếu sau đây:

1. Thu nhận dữ liệu và tiền xử lý.

2. Biểu diễn dữ liệu.

3. Nhận dạng, ra quyết định.

Bốn cách tiếp cận khác nhau trong lý thuyết nhận dạng là:

1 .Đối sánh mẫu dựa trên các đặc trưng được trích chọn.

2. Phân loại thống kê.

3. Đối sánh cấu trúc.

4. Phân loại dựa trên mạng nơ-ron nhân tạo.

Trong các ứng dụng rõ ràng là không thể chỉ dùng có một cách tiếp

cận đơn lẻ để phân loại “tối ưu” do vậy cần sử dụng cùng một lúc nhiều

phương pháp và cách tiếp cận khác nhau. Do vậy, các phương thức phân

loại tổ hợp hay được sử dụng khi nhận dạng và nay đã có những kết quả có

triển vọng dựa trên thiết kế các hệ thống lai (hybrid system) bao gồm nhiều

mô hình

kết hợp.

Việc giải quyết bài toán nhận dạng trong những ứng dụng mới, nảy

sinh trong cuộc sống không chỉ tạo ra những thách thức về thuật giải, mà

còn đặt ra những yêu cầu về tốc độ tính toán. Đặc điểm chung của tất cả

những ứng dụng đó là những đặc điểm đặc trưng cần thiết thường là nhiều,

không thể do chuyên gia đề xuất, mà phải được trích chọn dựa trên các thủ

tục phân tích dữ liệu.

### *1.2.7 Nén ảnh*

Nhằm giảm thiểu không gian lưu trữ. Thường được tiến hành theo cả hai cách khuynh hướng là nén có bảo toàn và không bảo toàn thông tin. Nén không bảo toàn thì thường có khả năng nén cao hơn nhưng khả năng phục hồi thì kém hơn. Trên cơ sở hai khuynh hướng, có 4 cách tiếp cận cơ bản trong nén ảnh:

• Nén ảnh thống kê: Kỹ thuật nén này dựa vào việc thống kê tần xuất

xuất hiện của giá trị các điểm ảnh, trên cơ sở đó mà có chiến lược

mã hóa thích hợp. Một ví dụ điển hình cho kỹ thuật mã hóa này

là \*.TIF

• Nén ảnh không gian: Kỹ thuật này dựa vào vị trí không gian của

các điểm ảnh để tiến hành mã hóa. Kỹ thuật lợi dụng sự giống nhau

của các điểm ảnh trong các vùng gần nhau. Ví dụ cho kỹ thuật này

là mã nén \*.PCX

• Nén ảnh sử dụng phép biến đổi: Đây là kỹ thuật tiếp cận theo

hướng nén không bảo toàn và do vậy, kỹ thuật thướng nến hiệu quả

hơn. \*.JPG chính là tiếp cận theo kỹ thuật nén này.

• Nén ảnh Fractal: Sử dụng tính chất Fractal của các đối tượng ảnh,

thể hiện sự lặp lại của các chi tiết. Kỹ thuật nén sẽ tính toán để chỉ

cần lưu trữ phần gốc ảnh và quy luật sinh ra ảnh theo nguyên lý

Fractal

## **1.3 Nhận dạng vật thể**

Object Detection đề cập đến khả năng của hệ thống máy tính và phần mềm để định vị các đối tượng trong một hình ảnh và xác định từng đối tượng. Object Detection đã được sử dụng rộng rãi để phát hiện khuôn mặt, phát hiện xe, đếm số người đi bộ, hệ thống bảo mật và xe không người lái. Có nhiều cách để nhận diện đối tượng có thể được sử dụng cũng như trong nhiều lĩnh vực thực hành.

Dù cho các công nghệ về nhận dạng và phân loại ảnh đã đạt được nhiều thành tựu đáng chú ý, lĩnh vực này vẫn phải đối mặt với nhiều vấn đề lớn về kỹ thuật cần giải quyết. Các vấn đề này thường bao gồm: sự méo của vật thể do môi trường có nhiễu, góc quay từ cảm biến hình ảnh tới vật thể. Đôi khi sự thay đổi của vật thể cần nhận dạng không được biểu diễn một cách chính xác do các giải thuật được ứng dụng với tập dữ liệu hạn chế. Ngoài ra, còn nhiều vấn đề này sinh trong thực tế làm cho vật cần nhận dạng bị méo trong quá trình xử lý hình ảnh. Trong các điều kiện thực tế khó khăn này, một hệ thống nhận dạng đáng tin cậy cần phải thực thi được chức năng nhận dạng và phân loại theo thời gian thực với tỉ lệ chuẩn xác cao. Do đó, việc cái tiến và phát triển các hệ thống xử lý ảnh cũng như các giải thuật là điều hết sức cần thiết đối với nhận dạng và phân loại vật thể cần sự chính xác và tốc độ cao.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# **CHƯƠNG 2: CƠ SỞ LÝ THUYẾT NHẬN DẠNG VẬT THỂ**

## **2.1 Phát hiện khuôn mặt (face recognition) và nhận diện khuôn mặt (face verification)**

Quá trình nhận diện khuôn mặt có thể được chia thành nhiều bước như hình bên dưới.

A white square with black text

Description automatically generated

1. **Face detection**: phát hiện một hay nhiều khuôn mặt trong ảnh. Việc này có thể thực hiện bằng nhiều cách ví dụ như **HaarCascade** hay dùng **Deep Learning** để phát hiện khuôn mặt.
2. **Feature extraction**: Trích xuất các đặc trưng quan trọng nahats từ ảnh chứa khuôn mặt (vùng ảnh chứa khuôn mặt được thực hiện sau bước 1). Có thể dùng FaceNet để trích xuất đặc trưng (128 dimensions).
3. **Face classification**: phân loại khuôn mặt dựa trên đặc trưng đã được trích xuất. Việc này có thể thực hiện bằng cách sử dụng **SVM** hay **Softmax**

Đây là pepline chung cho nhận diện khuôn mặt.

### 2.1.1 Face Recogniton

Có hình ảnh của người, hệ thống cần kiểm tra trong kho dữ liệu (có K người) xem ảnh này có trùng với ai trong cơ sở dữ liệu hay không, nếu trùng đó là ai (đưa ra ID chẳng hạn). Bài toán này phức tạp hơn hệ thống xác thực khuôn mặt. Ở đây phải so sánh ảnh 1:K. Chính vì điều này độ chính xác của face recognition thường thấp hơn face verification.

Nhận diện khuôn mặt có thể được ứng dụng trong việc chấm công, giám sát công dân…

Nhận dạng khuôn mặt liên quan đến việc lập bản đồ khuôn mặt của con người, không giống như xác minh khuôn mặt. Hay nói cách khác, đó là một quá trình nhận dạng sinh trắc học nhằm xác thực danh tính của một cá nhân bằng cách sử dụng các đặc điểm khuôn mặt của người đó. Camera chụp ảnh và quay video, chương trình bên trong lưu thông tin đó vào cơ sở dữ liệu. Nhận dạng khuôn mặt đối chiếu các mẫu sinh trắc học trên khuôn mặt của một cá nhân với cơ sở dữ liệu về các khuôn mặt đã biết để tìm ra kết quả khớp chính xác.

Sau đây là một số ngành dọc sử dụng nhận dạng khuôn mặt:

* Công ty
* Phòng thủ
* Liên bang
* An ninh và giám sát
* Sở cảnh sát

## 2.1.2 Face verification

Xác minh khuôn mặt thường bị nhầm lẫn với nhận dạng khuôn mặt vì cả hai đều yêu cầu khuôn mặt của bạn xác thực. Ví dụ: Bạn lên kế hoạch đi nghỉ ở một quốc gia khác, vì vậy bạn cần đăng nhập vào dịch vụ thị thực của chính phủ và bạn sử dụng thiết bị của mình để quét hộ chiếu làm bằng chứng nhận dạng nhằm đảm bảo rằng khuôn mặt của bạn khớp với hình ảnh của bạn trên hộ chiếu.

Xác minh khuôn mặt không chỉ giới hạn ở quy trình cấp thị thực mà còn được tận dụng rất nhiều. Về cơ bản, các lĩnh vực dưới đây yêu cầu xác minh khuôn mặt:

* Ngân hàng: Để mở tài khoản ngân hàng
* Các dịch vụ tài chính khác: Dành cho tín dụng, bảo hiểm, v.v.
* Chính phủ: Đối với đơn xin thị thực, giấy phép lái xe, hộ chiếu, v.v.
* Lĩnh vực giáo dục: Đăng ký vào một trường cao đẳng hoặc tham dự một kỳ thi
* Phương tiện truyền thông xã hội: Để thiết lập một tài khoản xã hội mới

Khi chúng ta có ID, có hình ảnh của người đó, nhập ID vào hệ thống rồi kiểm tra xem ảnh đó có phải là của người có mã ID kia không. Ở đây chỉ phải so sánh ảnh 1:1.

Trước hệ thống xác thực khuôn mặt đã có xác thực vân tay và mống mắt (chính xác hơn do ít thay đổi), khuôn mặt có thể thay đổi theo thời gian, make hay ánh sáng… Tuy nhiên hiện nay hệ thống xác thực bằng khuôn mặt cũng được ứng dụng rộng rãi.

## **2.2 Facenet và MTCNN**

### *2.2.1 MTCNN*

2.2.1.1 Khái niệm

MTCNN (Multi-task cascaded convolutional neural networks) - mạng NN nhằm phát hiện khuôn mặt và các facical landmarks (vị trí của mắt, mũi, miệng..) trong ảnh. MTCNN được công bố năm 2016 bởi Zhang et al.

A group of people smiling

Description automatically generated

MTCNN là một trong những công cụ phổ biến và chính xác nhất để phát hiện khuôn mặt hiện nay. MTCNN gồm 3 mạng NN kết nối với nhau theo kiểu tầng.

2.1.1.2 Kiến trúc của MTCNN

MTCNN bao gồm 3 mạng NN hay có thể gọi là 3 stages. Trong bước đầu tiên sử dụng mạng CNN nông (shallow) để nhanh chóng tạo ra các bounding boxes tiềm năng. Bước 2 tinh chỉnh lại bounding boxes nhận được từ bước 1 bằng mạng CNN phức tạp hơn. Bước cuối cùng sử dụng mạng CNN phức tạp hơn nữa để tinh chỉnh kết quả và đưa ra các facial landmarks.

A diagram of a layer

Description automatically generated with medium confidence

-Bước 1: The proposal network (P-Net)

Ở bước đầu tiên sẽ sử dụng mạng FCN (fully convolutional network). Mạng FCN khác mạng CNN ở chỗ mạng FCN không sử dụng lớp **Dense layer**. **P-Net** được sử dụng để có được các windows tiềm năng và bounding box regression vectors của chúng (tọa độ).

**Bounding box regression** là kỹ thuật để dự đoán vị trí của bounding box khi chúng ta cần phát hiện đối tượng (ở đây là khuôn mặt). Sau khi có được tọa độ của bounding boxes một vài tinh chỉnh được thực hiện để loại bỏ một số bounding boxes overlap với nhau (xem trong code sẽ có). Đầu ra của bước này là tất cả bounding boxes sau khi đã thực hiện sàng lọc.

- Bước 2: The Refine Network (R-Net)

Tất cả bounding boxes từ **P-Net** được đưa vào **R-Net**. Chú ý rằng **R-Net** là mạng CNN chứ không phải FCN. **R-Net** giảm số lượng bounding boxes xuống, tinh chỉnh lại tọa độ, có áp dụng Non-max suppression.

- Bước 3: The Output Network (O-Net)

Đầu ra của **R-Net** được sử dụng làm đầu vào của **O-Net**. Trong **O-Net** có đưa ra vị trí của facial landmarks (2 mắt, mũi, 2 vị trí của miệng). Như hình trên có thể thấy gần cuối chúng ta có 3 nhánh: xác suất xuất hiện khuôn mặt, tọa độ bounding box, tọa độ của facial landmarks (mỗi vị trí có tọa độ x và y, cái này ảnh hưởng đến số lượng units trong layer).

2.1.1.3 Mục đích của MTCNN

* Face classification

Đây là binary classification nên dùng loss function như sau: Bounding box regression

Đây là bài toán regression. Đối với mỗi window (bounding box) offset (độ lệch) giữa bounding box và ground truth bounding box được tính. Eucledian loss được áp dụng cho trường hợp này.

### Facial landmark localization

Xác định vị trí của facial landmarks. Đây cũng là bài toán regression. Có 5 landmraks: left eye, right eye, nose, left mounth corner, right mounth corners.

### 2.2.2 FaceNet

2.2.2.1 Khái niệm

FaceNet là mạng NN dùng để trích xuất đặc trưng từ ảnh mặt người (là mạng NN trong Siamese Network, Siamese Network bao gồm 2 mạng FaceNet giống hệt nhau). FaceNet được công bố năm 2015 bởi tác giả Schroff et al.

A computer screen shot of a computer screen

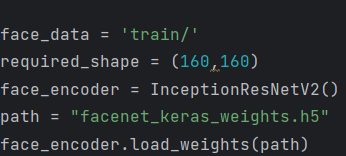
Description automatically generated

Đầu vào của FaceNet là ảnh khuôn mặt, đầu ra là là vector có chiều 128 thể hiện đặc trưng của khuôn mặt (còn gọn là encoding hay embedding). Gọi là **embedding** vì những thông tin quan trọng được **embedded** vào vector. Việc mapping dữ liệu có số chiều lớn (như ảnh chẳng hạn) xuống biểu diễn có số chiều nhỏ (embedding) là vấn đề rất hay thấy hiện nay.

FaceNet (chính xác Siamese Net - 2 mạng NN) đã được train dựa trên các bộ 3 ảnh (triplels). Vì vậy chúng ta có parameters sắn rồi nên mới có thể xác định được embedding.

Các giá trị trong embedding là kích thước của mắt, mũi, khoảng cách giữa 2 mắt… Các giá trị này rất quan trong, nhưng thực tế chúng ta không biết rõ ý nghĩa của chúng là gì.

Trong project này sử dụng pre-trained Keras FaceNet model  do Hiroki Taniai cung cấp . Trong project này, chúng em sử dụng  lệnh để chuyển đổi mô hình *Inception ResNet v2*  từ TensorFlow sang Keras. Nó được huấn luyện  trên tập dữ liệu MS-Celeb-1M và yêu cầu hình ảnh đầu vào có màu, giá trị pixel được làm trắng (được chuẩn hóa trên cả ba kênh) và có kích cỡ  160×160 pixel.



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# **CHƯƠNG 3: XÂY DỰNG GIAO DIỆN NHẬN DẠNG VẬT THỂ**

## **3.1 Công nghê sử dụng**

### ***3.1.1 HTML***

### 3.1.1.1 Khái niệm

HTML - HyperText Markup Language, là loại ngôn ngữ đánh dấu siêu văn bản, thường được sử dụng trong các tài liệu web.Về bản chất, HTML lại không được xem như một ngôn ngữ lập trình, mà nó giống như một ngôn ngữ xác định đâu là ý nghĩa, mục đích và cấu trúc của một tài liệu.

HTML dùng các thẻ để xác định các loại nội dung khác nhau, cũng như mục đích phục vụ cho trang web.

3.1.1.2 Mục đích

HTML có nhiệm vụ xây dựng cấu trúc siêu văn bản trên một website, hay khai báo các tập tin kỹ thuật số như hình ảnh, video, nhạc. Ngoài ra, nó còn được dùng để phân chia các đoạn văn bản, heading, links, blockquotes…

Đồng thời thực hiện dễ dang các công việc sau:

* Thêm tiêu đề, định dạng văn bản, ngắt dòng điều khiển.
* Tạo danh sách, nhấn mạnh văn bản, tạo ký tự đặc biệt, chèn hình ảnh và tạo liên kết.
* Góp phần xây dựng bảng, điều khiển một số kiểu mẫu.

### ***3.1.2 CSS***

3.1.2.1 Khái niệm

CSS - Cascading Style Sheets, là ngôn ngữ định dạng được sử dụng để mô tả cách trình bày trong các trang web, bao gồm màu sắc, cách bố trí và phông chữ, giúp nội dung thể hiện được phong cách riêng cho tổng thể website.

Nhờ có CSS nên các thành phần trong HTML của website có thể hiện thị bắt mắt trên giao diện của trang.

3.1.2.2 Phương thức hoạt động

CSS và HTML hoạt động hoàn toàn độc lập, đồng thời có thể sử dụng trên bất kỳ ngôn ngữ đánh dấu nào được xây dựng dựa trên XML.

CSS hoạt động tuân theo chuẩn chung mà W3C đưa ra.

Trong CSS được phân thành 3 phong cách chính, đó là:

* **Style CSS Internal:**Là style được tải lên mỗi khi trang web được refresh.
* **Style CSS Inline:** Ở phong cách này, bạn có thể dễ dàng chỉnh sửa bất kỳ yếu tố nào đó, mà không cần truy cập trực tiếp vào file CSS.
* **External Style:** Giúp bạn có thể tạo phong cách ở file khác áp dụng CSS vào trang bạn muốn, đồng thời cải thiện tối đa thời gian tải trang. Đặc biệt, bạn có thể ứng dụng phong cách này để tạo phong cách cho nhiều trang cùng thời điểm.

3.1.2.3 Mục đích

CSS sẽ giúp xây dựng nên “bộ mặt” hoàn hảo cho một website, đồng thời cũng mang đến cho nó phần giao diện “ưa nhìn” và trông lôi cuốn hơn trong quá trình giao tiếp. Ngoài ra, CSS còn giúp webmaster xác định được phong cách và định nghĩa nội dung web.

Thông qua đó, nhà lập trình có thể thực hiện nhanh chóng hơn các công việc sau:

* Tạo style và định dạng cho những nội dung được thể hiện dưới dạng văn bản của HTML.
* Tiết kiệm thời gian và công sức của lập trình viên nhờ khả năng điều khiển định dạng.
* Phân biệt cách hiển trị của trang web và nội dung của trang qua việc căn chỉnh bố cục, màu sắc cũng như font chữ.

### ***3.1.3 JavaScript***

3.1.3.1 Khái niệm

JavaScript hay còn được viết tắt là JS, đây là một trong những ngôn ngữ lập trình phổ biến được nhiều lập trình viên sử dụng, do Brendan Eich ra mắt vào năm 1995.

Nhờ các tính năng nổi bật của mình mà hầu hết các Slideshow, Pop-up quảng cáo, cùng tính năng Autocomplete của Google đều có sự góp sức của JavaScript.

3.1.3.2 Mục đích

JavaScript được sử dụng nhằm mục đích biến những website tĩnh nhàm chán sang trang thái động cuốn hút, đồng thời có khả năng tạo sự tương tác, cải thiện hiệu suất máy chủ và nâng cao trải nghiệm người dùng.Nhưng để thực hiện tốt những hoạt động trên, JavaScript phải cần đến sự hỗ trợ của HTML và CSS.

JavaScript còn được biết đến là một ngôn ngữ lập trình cho phép các nhà phát triển web thiết kế ra các trang web có khả năng tương tác.

Nhờ có sự hỗ trợ của JavaScript, mà các công việc trở nên đơn giản và dễ dàng hơn khi thực hiện như:

* Bắt đầu chỉnh sửa với thư viện ảnh, nhằm mục đích tạo nên một bố cục chuyển động.
* Hỗ trợ xây dựng các trò chơi, hoạt hình 2D hoặc 3D, ứng dụng cơ sở dữ liệu toàn diện để hoàn thiện website.
* Tăng cường các hành vi người dùng và kiểm soát mặc định trình duyệt

### ***3.1.4 API***

3.1.4.1 Khái niệm

API là cụm viết tắt của Giao diện chương trình ứng dụng. Trong ngữ cảnh API, từ “Ứng dụng” đề cập đến mọi phần mềm có chức năng riêng biệt. Giao diện có thể được xem là một hợp đồng dịch vụ giữa 2 ứng dụng. Hợp đồng này xác định cách thức hai ứng dụng giao tiếp với nhau thông qua các yêu cầu và phản hồi. Tài liệu API của các ứng dụng này chứa thông tin về cách nhà phát triển xây dựng cấu trúc cho những yêu cầu và phản hồi đó.

API là cơ chế cho phép 2 thành phần phần mềm giao tiếp với nhau bằng một tập hợp các định nghĩa và giao thức. Ví dụ: hệ thống phần mềm của cơ quan thời tiết chứa dữ liệu về thời tiết hàng ngày. Ứng dụng thời tiết trên điện thoại của bạn sẽ “trò chuyện” với hệ thống này qua API và hiển thị thông tin cập nhật về thời tiết hàng ngày trên điện thoại của bạn.

3.1.4.2 Phương thức hoạt động

Kiến trúc API thường được giải thích dưới dạng máy chủ và máy khách. Ứng dụng gửi yêu cầu được gọi là máy khách, còn ứng dụng gửi phản hồi được gọi là máy chủ. Như vậy, trong ví dụ về thời tiết, cơ sở dữ liệu của cơ quan thời tiết là máy chủ còn ứng dụng di động là máy khách.

API hoạt động theo 4 cách khác nhau, tùy vào thời điểm và lý do chúng được tạo ra.

* **API SOAP:** sử dụng Giao thức truy cập đối tượng đơn giản. Máy chủ và máy khách trao đổi thông đệp bằng XML. Đây là loại API kém linh hoạt được dùng phổ biến trước đây.
* **API RPC:** được gọi là Lệnh gọi thủ tục từ xa. Máy khách hoàn thành một hàm (hoặc thủ tục) trên máy chủ còn máy chủ gửi kết quả về cho máy khách.
* **API Websocket:** là một bản phát triển API web hiện đại khác sử dụng các đối tượng JSON để chuyển dữ liệu. API WebSocket hỗ trợ hoạt động giao tiếp hai chiều giữa ứng dụng máy khách và máy chủ. Máy chủ có thể gửi thông điệp gọi lại cho các máy khách được kết nối, điều này khiến loại API này hiệu quả hơn API REST.
* **API REST:** là loại API phổ biến và linh hoạt nhất trên web hiện nay. Máy khách gửi yêu cầu đến máy chủ dưới dạng dữ liệu. Máy chủ dùng dữ liệu đầu vào từ máy khách này để bắt đầu các hàm nội bộ và trả lại dữ liệu đầu ra cho máy khách. Hãy cùng xem xét API REST chi tiết hơn ở bên dưới.

### ***3.1.5 SpringBoot***

3.1.5.1 Khái niệm

Spring Boot là một dự án phát triển bởi ngôn ngữ Java trong hệ sinh thái Spring framework. Nó giúp cho các lập trình viên đơn giản hóa quá trình lập trình một ứng dụng với Spring, chỉ tập trung vào việc phát triển business cho ứng dụng. Spring Boot là một trong số những module của spring framework chuyên cung cấp các tính năng Rapid Application Development (RAD) để tạo ra và phát triển nhanh các ứng dụng độc lập dựa trên spring.



3.1.5.2 Các đặc tính cơ bản của SpringBoot

**SpringApplication:** SpringApplication là một class giúp khởi chạy các ứng dụng từ hàm main thuận tiện. Để bắt đầu ứng dụng, bạn chỉ cần gọi các method run.

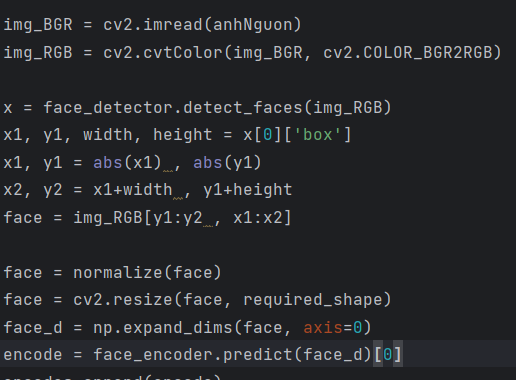
* **Profiles**: Spring Boot Profiles sẽ cung cấp một cách phân chia các cấu hình cho từng môi trường. Các annotation là *@Component* hoặc *@Configuration* có thể sẽ được đánh dấu profiles để giới hạn thời điểm hoặc môi trường sẽ được tải lên.
* **Externalized Configuration**: Externalized Configuration cho phép bạn có khả năng cấu hình được từ bên ngoài. Vì vậy, một ứng dụng được xây dựng có thể được vận hành và hoạt động trên nhiều môi trường khác nhau. Để thực hiện Externalized Configuration bạn có thể sử dụng các file properties, YAML, các tham số command line hay các biến môi trường.
* **Logging**: Tất cả các chức năng log nội bộ đều được spring boot sử dụng common logging. Chúng được quản lý một cách mặc định. Vì vậy, bạn không nên hoặc không cần sửa các dependency logging nếu các tuỳ biến customization không được yêu cầu.

### ***3.1.6 Thư viện openCV***

OpenCV là thư viện mã nguồn mở cho thị giác máy tính, xử lý ảnh và học máy, các tính năng tăng tốc GPU trong hoạt động thời gian thực. OpenCV được phát hành theo giấy phép BSD, do đó nó hoàn toàn phí cho cả học thuật và thương mại, cũng như được viết bằng C/C++ nên thư viện có thể tận dụng lợi thể xử lý đa lõi vậy nên rất phù hợp cho dự án này của nhóm trong việc nghiên cứu và phát triển khi ngồi trên ghế nhà trường.

Thư viện này sẽ được sử dụng cho việc đọc ảnh, chuyển đổi hệ màu với hàm cv2.cvtColor(img\_BGR, cv2.COLOR\_BGR2RGB) và resize ảnh với hàm cv2.resize(face, required\_shape) để xử lý cho ra ảnh phù hợp với model .

OpenCV có thể xử lý phát hiện các cạnh biên của ảnh.



### ***3.1.7 Thư viện Numpy***

Thư viện này có thể thực hiện các kỹ thuật ảnh đơn giản, như lật ảnh, trích xuất các tính năng và phân tích chúng.

Với ảnh màu là một mảng 3 chiều, bằng cách cắt mảng đa chiều, các kênh RGB có thể được tách ra.

Dưới đây là một số thao tác có thể được thực hiện bằng NumPy trên hình ảnh (hình ảnh được tải trong một biến có tên test\_img bằng cách sử dụng imread).

• Để lật hình ảnh theo hướng dọc, hãy sử dụng np.flipud (test\_img).

• Để lật hình ảnh theo hướng ngang, hãy sử dụng np.fliplr (test\_img).

• Để đảo ngược hình ảnh, hãy sử dụng test\_img [:: – 1] (hình ảnh sau khi lưu trữ

dưới dạng mảng numpy được đặt tên là <img\_name>).

• Và nhiều ứng dụng khác nữa

Ngoài ra Numpy cũng được sử dụng để làm việc với mảng số học trong ngữ cảnh của OpenCV, ví dụ mỗi hình ảnh được biểu diễn dưới dạng Numpy và các phép toán số học sẽ được thực hiện trên mảng này

Với những tiện ích như vậy thư viện Numpy sẽ giúp tối ưu hóa hiệu suất và cung cấp các công cụ hiệu quả đề xử lý các phép toán số học.

### **3.1.8 Các tiện ích khác**

. *datetime* : Được sử dụng để làm việc với ngày và giờ.

. *os*: thư viện hỗ trợ thao tác với hệ thống tệp và thư mục.

. *pickle* : Được sử dụng để tuần tự hóa và giải tuần tự hóa các đối tượng Python.

. *openpyxl*: thư viện hỗ trợ đọc và ghi file Excel.

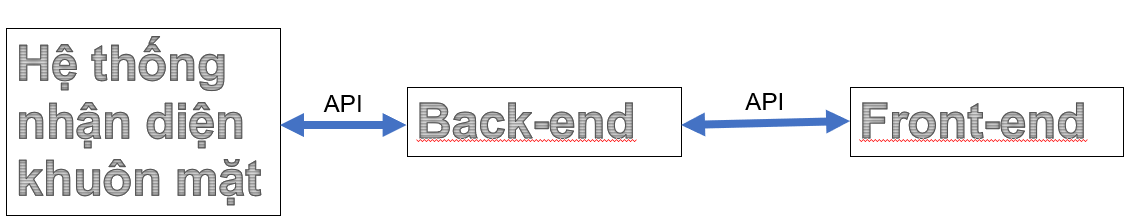
. *requests* : Được sử dụng để thực hiện các yêu cầu HTTP.

. *json* : Được sử dụng để làm việc với dữ liệu JSON.

. *base64* : Dùng để mã hóa và giải mã dữ liệu theo định dạng base64.

## **3.2 Kiến trúc hiện thống**

* Kiến trúc hệ thống được chia thành 3 phần chính.



* Front-end:

1. Được dùng để truyền ảnh về để xử lý và hiển thị cho người dùng kết quả.
2. Các ngôn ngữ được sử dụng là: HTML, CSS, Javascipt

* Back-end:

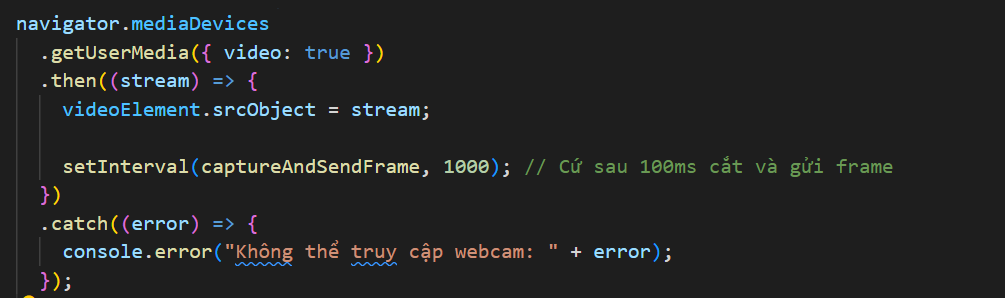
1. Nơi trung gian giữa hệ thống nhận diện khuôn mặt và front-end
2. Các ngôn ngữ được sử dụng là: Framework Spring

* Hệ thống nhận diện khuôn mặt:

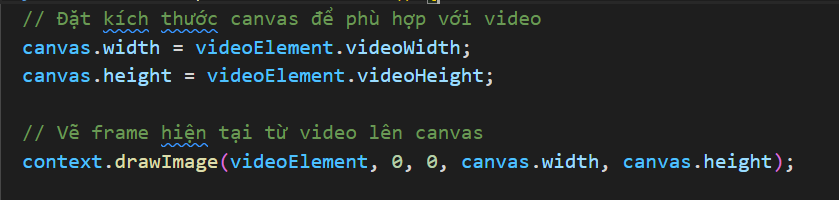
1. Được dùng để xử lý ảnh từ back-end gửi về và gửi ngược lại khi xử lý xong
2. Các ngôn ngữ, thư viện được sử dụng là: Python, Mtcnn, Facenet

## **3.3 Quá trình chạy của hệ thống**

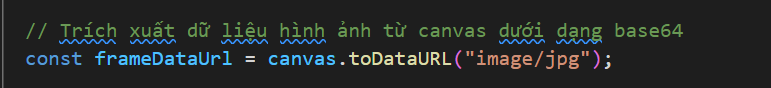
1. Đầu tiên, từ front-end bật camera để lấy dữ liệu hình ảnh video qua đoạn script



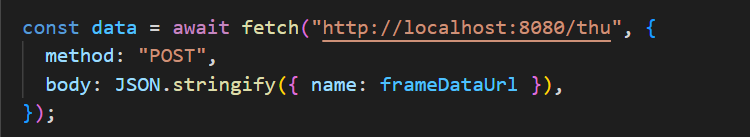
1. Sau khi truy cập được camera thành công, video từ gian thực được chính camera ghi lại sẽ được cắt thành từ frame ảnh:



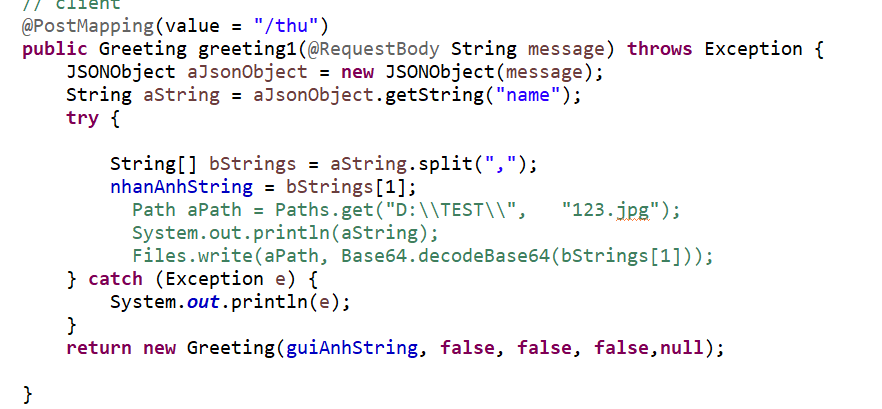
1. Sau khi lấy được frame ảnh hiện tại từ camera thì tiếp tục, hình ảnh được mã hóa dưới dạng base64:



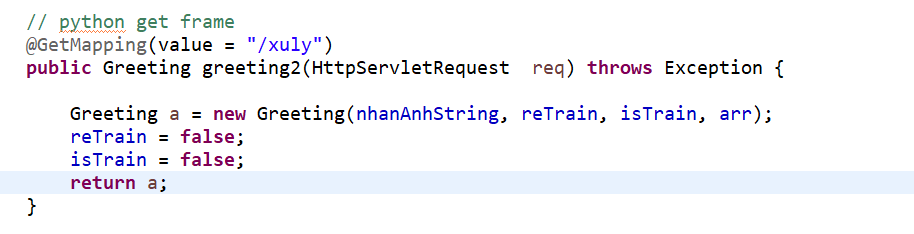
1. Tiếp đó, hình ảnh được hóa quá dưới dạng base64 và được chuyển thành json và gửi cho back-end qua API:



1. Ở bên phía back-end với url <http://localhost:8080/thu> sẽ nhận được ảnh qua phương thức:



1. Bên back-end sẽ nhận được một request chính là một chuỗi json được gửi từ front-end có tên biến là message thì sẽ được chuyển thành một Object
2. Tiếp theo đó, ảnh được mã hóa bằng base64 sẽ được lấy qua thuộc tính name mà trước đó đã được cho ảnh. Sau khi được xử lý chuỗi thì sẽ được gắn chuỗi đó qua tên nhanAnhString
3. Cuối cùng, back-end sẽ gửi ảnh cho hệ thống nhận diện khuôn mặt qua thương thức:



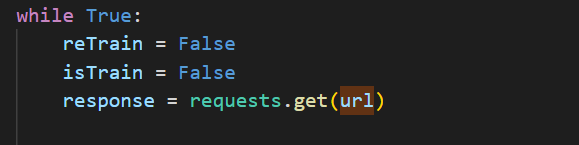
( nhanAnhString là ảnh được mã hóa thành base64,

reTrain: là một biến để cho hệ thống nhận diện khuôn mặt biết có mệnh lệnh train hay không,

isTrain: là một biến để giúp hệ thống nhận diện khuôn mặt biết có mệnh lệnh khi có sự thay đổi khi người dụng chọn 1 model mới,

arr: là một mảng chứa các id của người dùng khi được chọn để train)

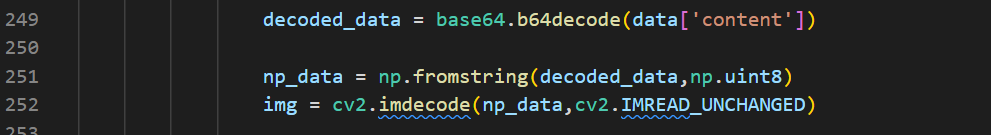
1. Sau khi hệ thống nhận diện khuôn mặt lấy được một chuỗi json được gửi về từ back-end bằng cách gọi hàm request.get(url) (url: <http://localhost:8080/xuly>)



1. Tiếp đó, hệ thống nhận diện khuôn mặt sẽ chuyển chuỗi json thành một Object để xử lý:



1. Sau khi lấy được ảnh được mã hóa bởi base64, hệ thống thông minh sẽ giải mã base64 thành ảnh để thực hiện quá trình nhận diện khuôn mặt (data[‘content’]) là chuỗi mã hóa của ảnh từ back-end gửi sang và dùng cv2 để giải mã và gán vào biên img



1. Tiếp theo, hệ thống nhận diện khuôn mặt sẽ gọi đế hàm detect để thực hiện quá trình nhận diện khuôn mặt



(img: là ảnh sau khi được ảnh mã

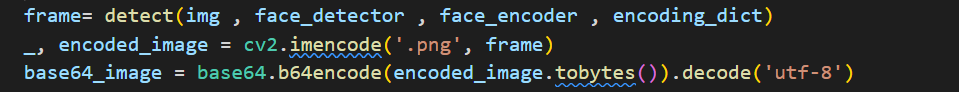
face\_detector: là một biến được gán bằng thư viện mtcnn 

face\_encoder: là một model của facenet

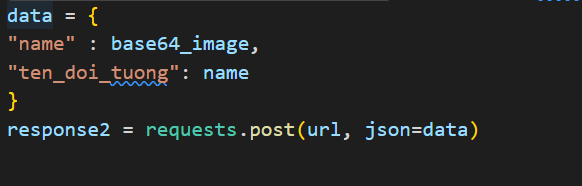


encoding\_dict: là một model đã được train từ dữ liệu mà người dùng chọn trước đó)

1. Sau khi hàm detect thực hiện công việc nhận diện xong sẽ trả về một ảnh đã được vẽ khung và tên lên trên khuôn mặt đó.
2. Sau đó, ảnh sẽ được mã hóa thành base 64



1. Tiếp đó, khởi tạo một biến có kiểu dữ liệu là dict và chuyển biến đó thành json và gửi lên back-end:

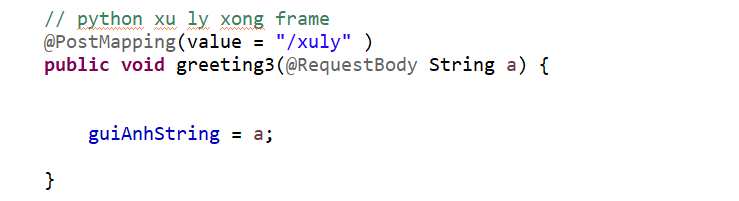


(base64\_img: là ảnh sau khi được nhận diện và mã hóa thành base 64,

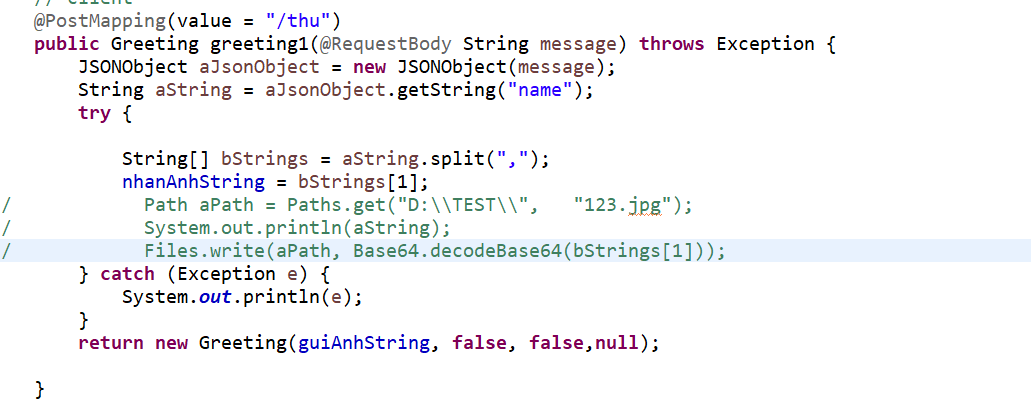
Name: là tên của đối tưởng ảnh khi được nhận diện

url: <http://localhost:8080/xuly>)

1. Ở phía back-end, khi nhận được json và sẽ được lưu vào biến guiAnhString qua phương thức



1. Tiếp đến, back-end sẽ trả lại cho front-end một Object qua phương thức:



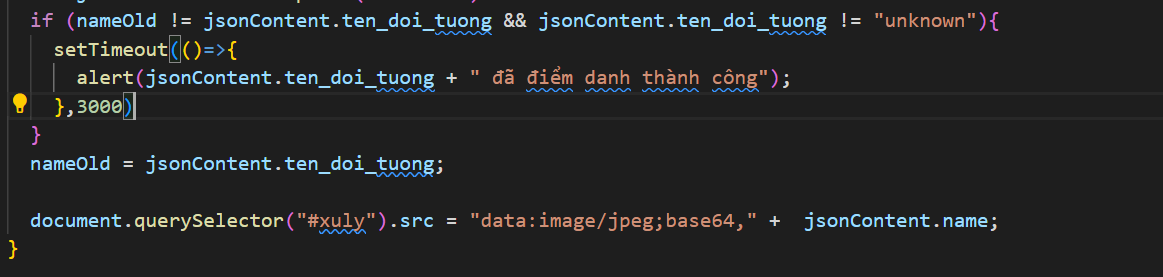
(guiAnhString chính là chuỗi json mà Hệ thống nhận diện khuôn mặt gửi cho bên back-end)

1. Ở phía front-end, sẽ nhận được chuỗi json của back-end trả về qua đoạn script



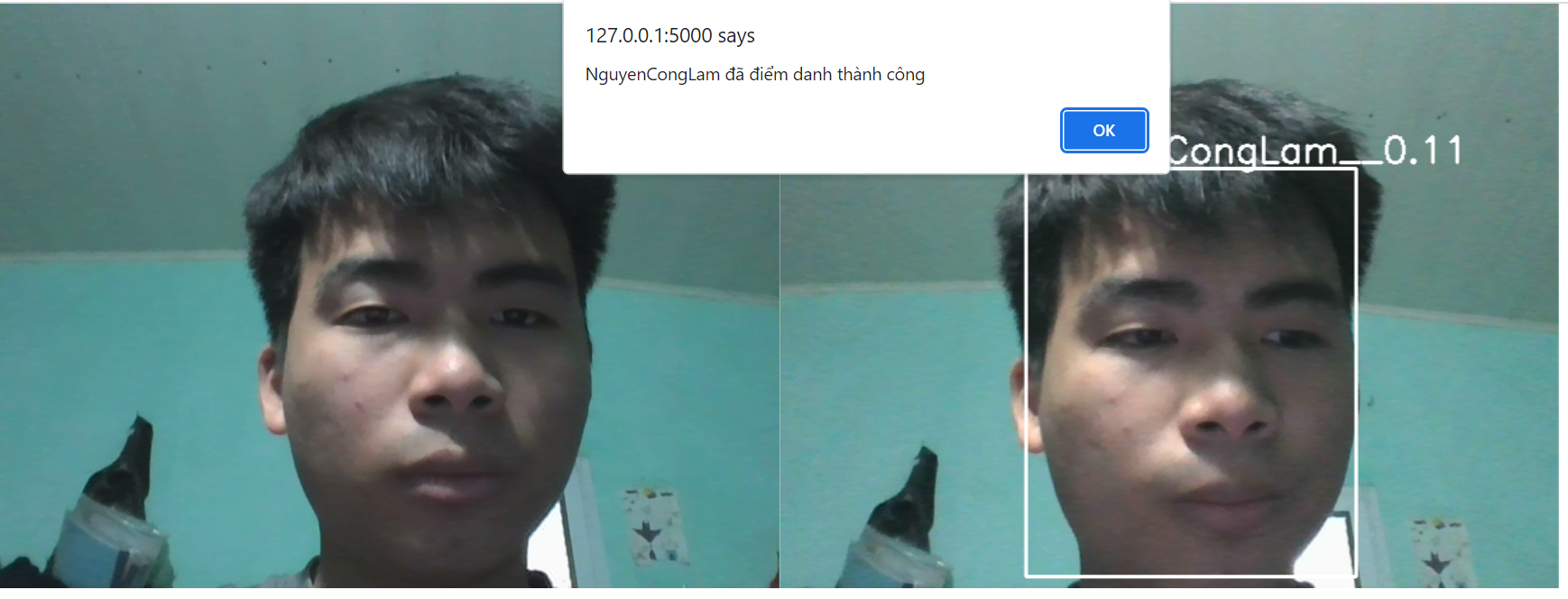
(a.content: chính là chuỗi json mà Hệ thống nhận diện khuôn mặt gửi cho bên back-end vì đây vẫn chỉ là chuỗi json lên phải chuyển thành Object bằng JSON.parse)

1. Sau khi đã chuyển được thành Object thì front-end sẽ hiển thị ảnh của hệ thống nhận diện khuôn mặt xử lý và tên của ảnh để hiện thông báo điểm danh thành công cho đối tưởng)



**3.4 Kết quả khi chạy chương trình**

1. Giao diện nhận diện khuôn mặt



2. Giao diện để thêm mẫu ( Thêm ảnh có khuôn mặt lên cho ảnh chỉ có 1 mặt, không cần phải thực hiện quá trình cắt ảnh)

B1: Giao diện chính



B2: Chọn Quản lý mẫu ( để thêm mẫu)



B3: Chọn face recognition. Giao diện hiện ra



B4: Nhập tên và chon tìm kiếm giao diện hiện ra



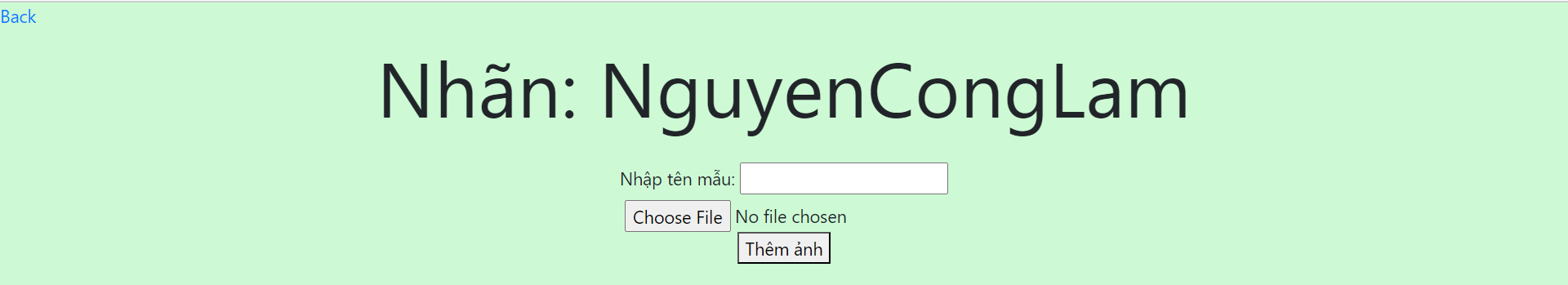
B5: Chọn xem chi tiết( bất kỳ 1 dòng nào) giao diện hiện ra



B6: Tìm kiếm vào chọn tìm kiếm và bấm xem chi tiết để xem mẫu( ảnh) đã có



B7: Chọn vào thêm mẫu giao diện hiện ra



B8: Upload ảnh vào chọn thêm( Hoàn thành thêm mẫu)

3. Giao diện train model

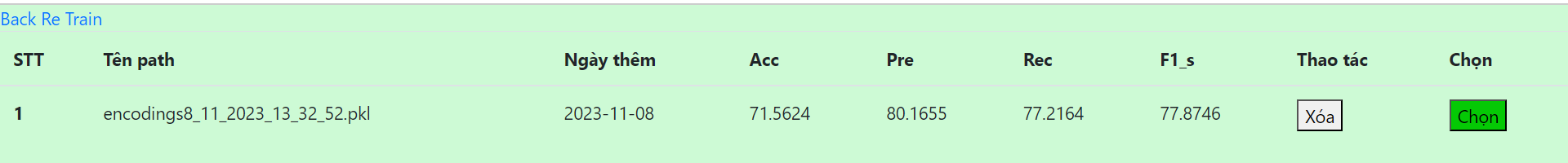
B1: Giao diện chính



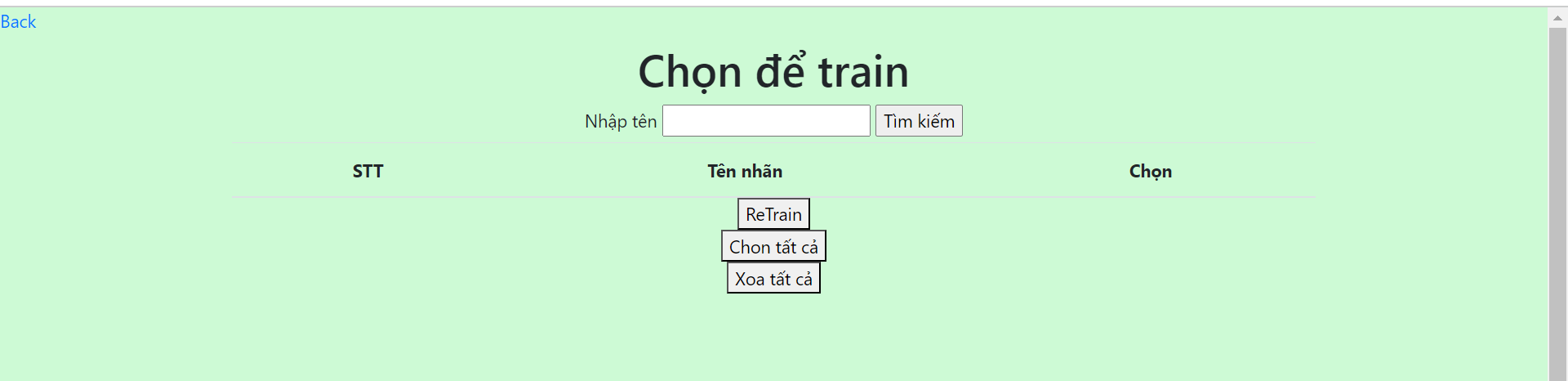
B2: Chọn Quản lý model( để train)



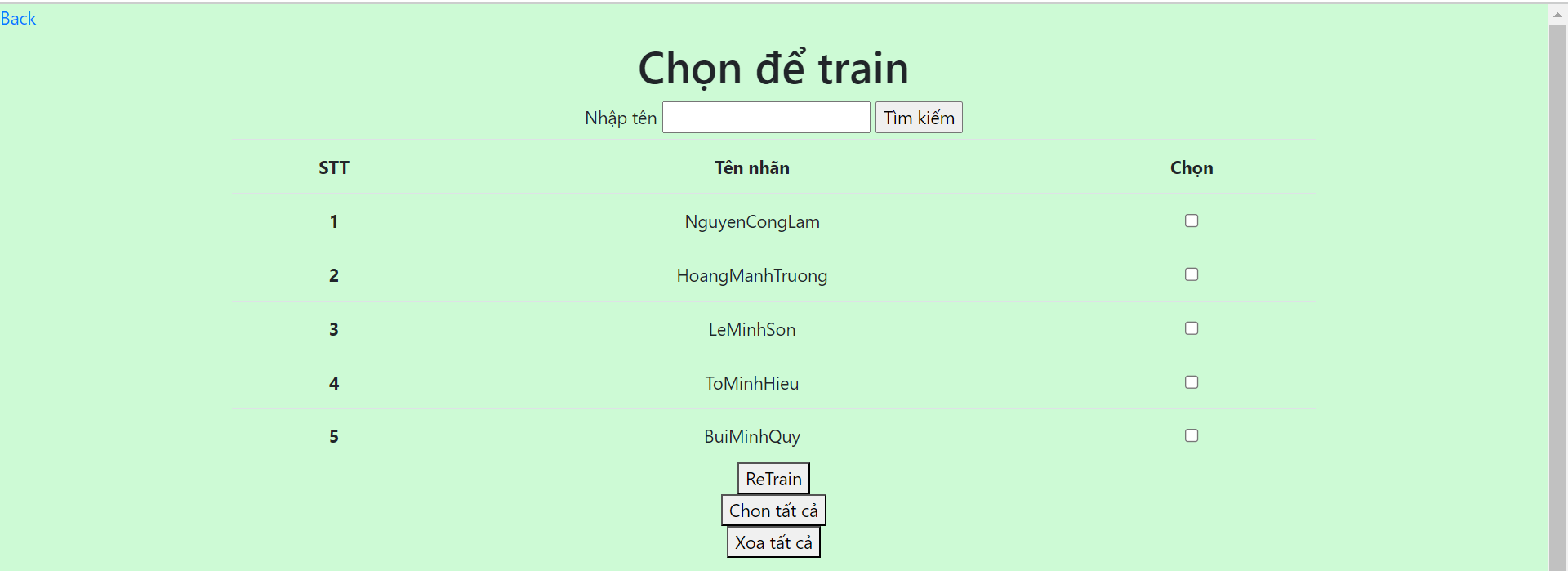
B3: Chọn face recogniton giao diện hiện ra



B4: Chọn Re train giao diện hiện ra



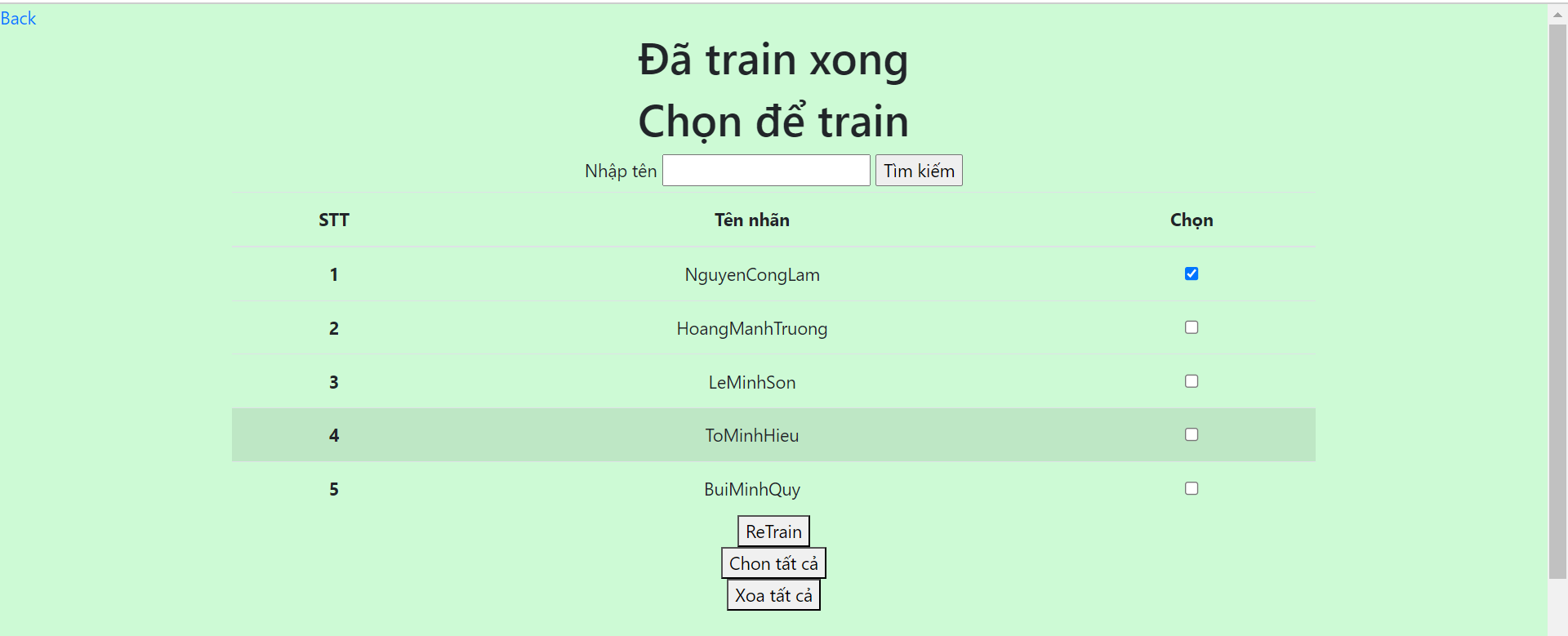
B5: Nhập tên (nhãn) và bấm tìm kiếm để chọn tên



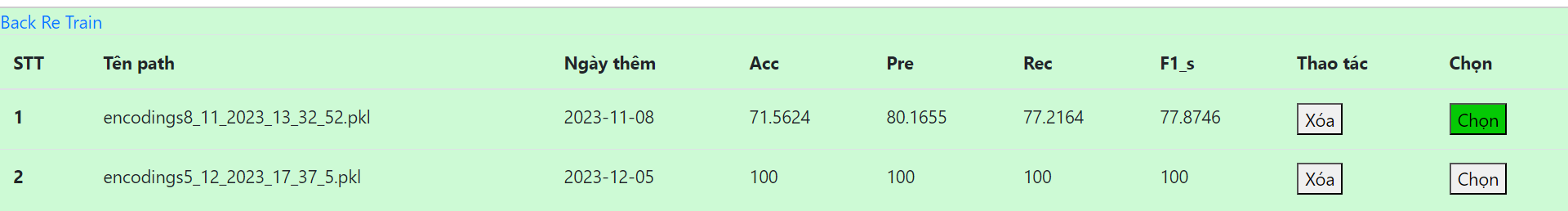
B6: Chọn vào các ô để train và bấm ReTrain ( Sẽ có thông báo đang train)



B7: Giao diện khi train xong(Sẽ có thông báo đã train xong)

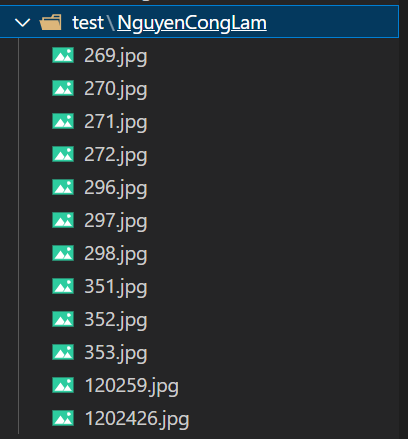


B8: Bấm vào back để quay lại giao diện để chọn model



( Sau khi train xong model đó sẽ tự đánh giá qua 4 Accuracy, Precision, Recall, F1-score)

( Trong thư mục test trong Fordel HeThongNhanDienKhuonMat phải để đúng tên người muốn train thì mới có thể đánh giá chính xác nhất)



Kết quả nhận diện được ghi trong file appending\_values.xlsx