TRƯỜNG ĐẠI HỌC ĐIỆN LỰC

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO MÔN HỌC  
ĐỒ ÁN LẬP TRÌNH WEB**

**ĐỀ TÀI**

Xây dựng một ứng dụng login bằng đối sánh khuôn mặt trên nền Web

**Sinh viên thực hiện : Lê Văn Vững**

**Giảng viên hướng dẫn : Ngô Hoàng Huy**

**Ngành : CÔNG NGHỆ THÔNG TIN**

**Chuyên ngành : TTNT&TGMT**

**Lớp : D14TTNT&TGMT**

**Niên Khóa : 2019-2023**

**Hà Nội, tháng 06 năm 2020**

**PHIẾU CHẤM ĐIỂM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Họ và Tên sinh viên** | **Nội dung thực hiện** | **Điểm** | **Chữ ký** |
| 1 | Lê Văn Vững |  |  |  |

|  |  |  |
| --- | --- | --- |
| **Giảng viên chấm thi** | **Chữ ký** | **Ghi chú** |
| Giảng viên chấm thi 1: |  |  |
| Giảng viên chấm thi 2: |  |  |

**MỤC LỤC**

[LỜI MỞ ĐẦU 4](#_Toc106489103)

[CHƯƠNG 1 : GIỚI THIỆU ĐỀ TÀI 5](#_Toc106489104)

[Giới thiệu 5](#_Toc106489105)

[CHƯƠNG 2 : Xây Dựng Cơ Sở Dữ liệu 6](#_Toc106489106)

[2.1 Sơ đồ tổng quan hệ thống 6](#_Toc106489107)

[2.2 Diagram 7](#_Toc106489108)

[2.3 USE CASE 8](#_Toc106489109)

[CHƯƠNG 3: Công Nghệ Sử Dụng 8](#_Toc106489110)

[3.1. Giới thiệu về công nghệ sử dụng 8](#_Toc106489111)

[3.1.1.Thư viện Open CV 9](#_Toc106489112)

[3.1.2.Thư viện TensorFlow 2 10](#_Toc106489113)

[3.1.3. Thư viện Scikit-learn 11](#_Toc106489114)

[3.1.5.Flask 12](#_Toc106489115)

[3.2. Thuật toán CNN 13](#_Toc106489116)

[3.3.Các bước tiến hành 16](#_Toc106489117)

[3.4.Mô tả dự án 17](#_Toc106489118)

[CHƯƠNG 4: Cài đặt và kiểm thử 18](#_Toc106489119)

[4.1.Triển khai bài toán 18](#_Toc106489120)

[4.1.1.Chuẩn bị dữ liệu 18](#_Toc106489121)

[4.1.2.Train mô hình 18](#_Toc106489122)

[4.1.3. kết quả dự đoán 20](#_Toc106489123)

[4.1.4.trang web face login 21](#_Toc106489124)

[KẾT LUẬN 21](#_Toc106489125)

[DANH MỤC TÀI LIỆU THAM KHẢO 22](#_Toc106489126)

# LỜI MỞ ĐẦU

Công nghệ thông tin (CNTT) ngày càng có vai trò quan trọng trong cuộc sống hằng ngày của chúng ta. Việc ứng dụng CNTT vào các lĩnh vực trong đời sống giúp công việc được tiến hành nhanh chóng và hiệu quả hơn. Có rất nhiều công việc mới phát triển song song với sự phát triển của CNTT, một trong những số đó là phân tích chuẩn đoán dữ liệu – big data, hướng đi mang lại hiệu quả rất lớn.

Nhóm chúng em chọn đề tài “ Xây dựng một ứng dụng login bằng đối sánh khuôn mặt trên nền Web” nhằm tìm hiểu sâu hơn về dữ liệu cách khai thác và sử dụng hiệu quả, từ đó viết một ứng dụng cụ thể thử nghiệm làm cơ sở củng cố kiến thức và định hướng, kế hoạch xây dựng các ứng dụng trong tương lai.

# CHƯƠNG 1 : GIỚI THIỆU ĐỀ TÀI

## Giới thiệu

Nhận dạng khuôn mặt hiện nay có rất nhiều hướng nghiên cứu nhằm nâng cao hiệu suất nhận dạng và tối ưu hóa hệ thống. Bài báo này nghiên cứu và phát triển hai hệ thống nhận dạng khuôn mặt đơn giản theo hai phương pháp đối sánh và mạng nơron cùng dựa trên đặc trưng PCA. Hệ thống được kiểm tra trên cơ sở dữ liệu ORL của AT&T. Kết quả thực nghiệm được phân tích để đánh giá hiệu suất nhận dạng và tính ổn định của hai hệ thống nhận dạng trong các điều kiện kiểm thử khác nhau. Kết quả cũng chỉ ra hiệu suất vượt trội của mạng nơron so với phương pháp đối sánh.

Nhận dạng khuôn mặt người là một công nghệ được ứng dụng rộng rãi trong đời sống hằng ngày của con người như các hệ thống giám sát, quản lý vào ra, tìm kiếm thông tin một người nổi tiếng,…Có rất nhiều phương pháp nhận dạng khuôn mặt để nâng cao hiệu suất tuy nhiên dù ít hay nhiều những phương pháp này đang vấp phải những thử thách về độ sáng, hướng 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 [1][2]. Trong đó, PCA là phương pháp trích rút đặc trưng nhằm giảm số chiều của ảnh tuy đơn giản nhưng mang lại hiệu quả tốt. Nhận dạng khuôn mặt dùng PCA kết hợp với mạng nơron là phương pháp mang lại hiệu quả nhận dạng cao bởi nó phát huy được ưu điểm của PCA và mạng nơron [3]. Hệ thống hoạt động ổn định và có tính thích nghi cao khi dữ liệu đầu vào thay đổi nhiều.

# CHƯƠNG 2 : Xây Dựng Cơ Sở Dữ liệu

## 2.1 Sơ đồ tổng quan hệ thống

Diagram

Description automatically generated

## 2.2 Diagram

Diagram

Description automatically generated

## 2.3 USE CASE

A picture containing text, indoor, map

Description automatically generated

# CHƯƠNG 3: Công Nghệ Sử Dụng

## 3.1. Giới thiệu về công nghệ sử dụng

Chúng tôi đã triển khai trang đăng nhập ứng dụng web Flask bao gồm xác minh khuôn mặt (1-1 để xác minh xem người đăng nhập có thực sự là người đó hay không), vì mục đích bảo mật, với cơ chế phát hiện trực tiếp (để kiểm tra xem người được phát hiện trên máy ảnh là người THẬT hay GIẢ (ví dụ: hình ảnh, video, v.v. của người đó)), để Chống giả mạo (Người khác giả danh là người đó), được xây dựng bằng Mạng thần kinh kết hợp. Sau trang đăng nhập, chúng tôi cũng cung cấp trình giữ chỗ trang web để sử dụng trong tương lai.

Kết quả

\* Nhận dạng khuôn mặt hoạt động tốt để phát hiện khuôn mặt và nhận dạng chính xác khuôn mặt.

\* Tính năng phát hiện độ sống động hoạt động tốt với việc phân loại hình ảnh và video giả từ việc giả mạo trên điện thoại thông minh.

\* Tính năng phát hiện độ sống động cũng được đào tạo để có thể phân loại hình ảnh in rắn (hình ảnh trên giấy và thẻ). Nhưng nó chỉ được đào tạo với khoảng 10 hình ảnh, vì vậy nó không hoạt động tốt mọi lúc (đọc phần "Sử dụng toàn bộ quy trình làm việc và đào tạo mô hình của riêng bạn", trong trường hợp bạn muốn tự đào tạo với tập dữ liệu lớn hơn)

Thư viện sử dụng :

- OpenCV

- TensorFlow 2

- Scikit-learn

- Face\_recognition

- Flask

IDE thử nghiệm : Visual Studio code,Pycharm

Công cụ lưu trữ:

- SQLite

- SQLAlchemy (for Flask)

## 3.1.1.Thư viện Open CV

Chính thức ra mắt vào năm 1999, dự án OpenCV ban đầu là một sáng kiến của Intel Research nhằm thúc đẩy các ứng dụng chuyên sâu về CPU , một phần của chuỗi dự án bao gồm theo dõi tia thời gian thực và các bức tường hiển thị 3D . [4] Những người đóng góp chính cho dự án bao gồm một số chuyên gia tối ưu hóa ở Intel Nga, cũng như Nhóm Thư viện Hiệu suất của Intel. Trong những ngày đầu của OpenCV, các mục tiêu của dự án được mô tả [5] là:

Nghiên cứu tầm nhìn trước bằng cách cung cấp mã không chỉ mở mà còn được tối ưu hóa cho cơ sở hạ tầng tầm nhìn cơ bản. Không cần phải phát minh lại bánh xe .

Phổ biến kiến thức về tầm nhìn bằng cách cung cấp một cơ sở hạ tầng chung mà các nhà phát triển có thể xây dựng trên đó, để mã sẽ dễ đọc và dễ chuyển giao hơn.

Nâng cao các ứng dụng thương mại dựa trên tầm nhìn trước bằng cách cung cấp miễn phí mã di động , tối ưu hóa hiệu suất - với giấy phép không yêu cầu mã phải mở hoặc tự do.

Phiên bản alpha đầu tiên của OpenCV đã được phát hành cho công chúng tại Hội nghị IEEE về Thị giác Máy tính và Nhận dạng Mẫu vào năm 2000, và 5 bản beta đã được phát hành từ năm 2001 đến 2005. Phiên bản 1.0 đầu tiên được phát hành vào năm 2006. Bản phát hành trước phiên bản 1.1 " "được phát hành vào tháng 10 năm 2008.

Bản phát hành chính thứ hai của OpenCV là vào tháng 10 năm 2009. OpenCV 2 bao gồm những thay đổi lớn đối với giao diện C ++ , hướng đến các mẫu an toàn hơn, dễ dàng hơn, các chức năng mới và triển khai tốt hơn cho các mẫu hiện có về mặt hiệu suất (đặc biệt là trên đa hệ thống cốt lõi). Các bản phát hành chính thức hiện diễn ra sáu tháng một lần [6] và việc phát triển hiện được thực hiện bởi một nhóm độc lập của Nga được hỗ trợ bởi các tập đoàn thương mại.

Vào tháng 8 năm 2012, hỗ trợ cho OpenCV đã được tổ chức phi lợi nhuận OpenCV.org tiếp quản, tổ chức này duy trì một nhà phát triển [7] và trang web của người dùng. [số 8]

Vào tháng 5 năm 2016, Intel đã ký một thỏa thuận để mua lại Itseez, [9] nhà phát triển hàng đầu của OpenCV. [10]

Vào tháng 7 năm 2020, OpenCV đã công bố và bắt đầu chiến dịch Kickstarter cho OpenCV AI Kit , một loạt các mô-đun phần cứng và phần bổ sung cho OpenCV hỗ trợ Spatial AI.

## 3.1.2.Thư viện TensorFlow 2

TensorFlow là hệ thống thế hệ thứ hai của Google Brain. Phiên bản 1.0.0 được phát hành vào ngày 11 tháng 2 năm 2017. [16] Trong khi triển khai tham chiếu chạy trên các thiết bị đơn lẻ, TensorFlow có thể chạy trên nhiều CPU và GPU (với các phần mở rộng CUDA và SYCL tùy chọn cho tính toán mục đích chung trên các đơn vị xử lý đồ họa ). [17] TensorFlow có trên nền tảng Linux , macOS , Windows và điện toán di động 64-bit bao gồm Android và iOS .

Kiến trúc linh hoạt của nó cho phép dễ dàng triển khai tính toán trên nhiều nền tảng (CPU, GPU, TPU ) và từ máy tính để bàn đến cụm máy chủ đến thiết bị di động và thiết bị cạnh.

Các tính toán TensorFlow được biểu thị dưới dạng biểu đồ luồng dữ liệu trạng thái . Tên TensorFlow bắt nguồn từ các hoạt động mà các mạng nơ-ron như vậy thực hiện trên các mảng dữ liệu đa chiều, được gọi là tensor . Trong Hội nghị Google I / O vào tháng 6 năm 2016, Jeff Dean tuyên bố rằng 1.500 kho lưu trữ trên GitHub đã đề cập đến TensorFlow, trong đó chỉ có 5 kho của Google. [18]

Vào tháng 12 năm 2017, các nhà phát triển từ Google, Cisco, RedHat, CoreOS và CaiCloud đã giới thiệu Kubeflow tại một hội nghị. Kubeflow cho phép vận hành và triển khai TensorFlow trên Kubernetes .

Vào tháng 3 năm 2018, Google đã công bố TensorFlow.js phiên bản 1.0 cho máy học bằng JavaScript . [19]

Vào tháng 1 năm 2019, Google đã công bố TensorFlow 2.0. [20] Nó chính thức có sẵn vào tháng 9 năm 2019. [10]

Vào tháng 5 năm 2019, Google đã công bố Đồ họa TensorFlow để học sâu về đồ họa máy tính. [21]

## 3.1.3. Thư viện Scikit-learn

Scikit-learn (Sklearn) là thư viện mạnh mẽ nhất dành cho các thuật toán học máy được viết trên ngôn ngữ Python. Thư viện cung cấp một tập các công cụ xử lý các bài toán machine learning và statistical modeling gồm: classification, regression, clustering, và dimensionality reduction.

Thư viện được cấp phép bản quyền chuẩn FreeBSD và chạy được trên nhiều nền tảng Linux. Scikit-learn được sử dụng như một tài liệu để học tập.

3.1.4.Thư viện Face\_recognition

Nhiều người đã quen thuộc với công nghệ nhận dạng khuôn mặt thông qua FaceID được sử dụng để mở khóa iPhone (tuy nhiên, đây chỉ là một ứng dụng nhận dạng khuôn mặt). Thông thường, nhận dạng khuôn mặt không dựa vào cơ sở dữ liệu khổng lồ về ảnh để xác định danh tính của một cá nhân - nó chỉ xác định và công nhận một người là chủ sở hữu duy nhất của thiết bị, đồng thời hạn chế quyền truy cập của những người khác.

Ngoài việc mở khóa điện thoại, tính năng nhận dạng khuôn mặt hoạt động bằng cách khớp khuôn mặt của những người đi ngang qua máy ảnh đặc biệt với hình ảnh của những người trong danh sách theo dõi. Danh sách theo dõi có thể chứa hình ảnh của bất kỳ ai, kể cả những người không bị nghi ngờ có hành vi sai trái nào và hình ảnh có thể đến từ bất kỳ đâu - ngay cả từ các tài khoản mạng xã hội của chúng tôi. Các hệ thống công nghệ trên khuôn mặt có thể khác nhau, nhưng nhìn chung, chúng có xu hướng hoạt động như sau:

Bước 1: Nhận diện khuôn mặt

Máy ảnh phát hiện và định vị hình ảnh của một khuôn mặt, một mình hoặc trong một đám đông. Hình ảnh có thể cho thấy người đó đang nhìn thẳng về phía trước hoặc trong hồ sơ.

Bước 2: Phân tích khuôn mặt

Tiếp theo, một hình ảnh của khuôn mặt được chụp và phân tích. Hầu hết công nghệ nhận dạng khuôn mặt dựa trên hình ảnh 2D thay vì 3D vì nó có thể khớp hình ảnh 2D với ảnh công khai hoặc ảnh trong cơ sở dữ liệu thuận tiện hơn. Phần mềm đọc hình dạng khuôn mặt của bạn. Các yếu tố chính bao gồm khoảng cách giữa hai mắt, độ sâu của hốc mắt, khoảng cách từ trán đến cằm, hình dạng của gò má và đường viền của môi, tai và cằm. Mục đích là để xác định các điểm mốc trên khuôn mặt là chìa khóa để phân biệt khuôn mặt của bạn.

Bước 3: Chuyển đổi hình ảnh thành dữ liệu

Quá trình chụp khuôn mặt chuyển đổi thông tin tương tự (khuôn mặt) thành một tập hợp thông tin kỹ thuật số (dữ liệu) dựa trên các đặc điểm trên khuôn mặt của người đó. Phân tích khuôn mặt của bạn về cơ bản được biến thành một công thức toán học. Mã số được gọi là khuôn mặt. Giống như cách mà dấu vân tay là duy nhất, mỗi người có khuôn mặt của riêng họ.

Bước 4: Tìm một kết quả phù hợp

Sau đó, khuôn mặt của bạn được so sánh với cơ sở dữ liệu của các khuôn mặt đã biết khác. Ví dụ, FBI có quyền truy cập lên tới 650 triệu bức ảnh , được lấy từ nhiều cơ sở dữ liệu của tiểu bang. Trên Facebook, bất kỳ ảnh nào được gắn thẻ tên người đều trở thành một phần trong cơ sở dữ liệu của Facebook, cơ sở dữ liệu này cũng có thể được sử dụng để nhận dạng khuôn mặt. Nếu khuôn mặt của bạn khớp với một hình ảnh trong cơ sở dữ liệu nhận dạng khuôn mặt, thì việc xác định sẽ được thực hiện.

Trong tất cả các phép đo sinh trắc học, nhận dạng khuôn mặt được coi là tự nhiên nhất. Theo trực giác, điều này có ý nghĩa, vì chúng ta thường nhận ra bản thân và những người khác bằng cách nhìn vào khuôn mặt, thay vì dấu ngón tay cái và mống mắt. Người ta ước tính rằng hơn một nửa dân số thế giới thường xuyên tiếp xúc với công nghệ nhận dạng khuôn mặt.

## 3.1.5.Flask

Flask là một web frameworks, nó thuộc loại micro-framework được xây dựng bằng ngôn ngữ lập trình Python. Flask cho phép bạn xây dựng các ứng dụng web từ đơn giản tới phức tạp. Nó có thể xây dựng các api nhỏ, ứng dụng web chẳng hạn như các trang web, blog, trang wiki hoặc một website dựa theo thời gian hay thậm chí là một trang web thương mại. Flask cung cấp cho bạn công cụ, các thư viện và các công nghệ hỗ trợ bạn làm những công việc trên.

Flask là một micro-framework. Điều này có nghĩa Flask là một môi trường độc lập, ít sử dụng các thư viện khác bên ngoài. Do vậy, Flask có ưu điểm là nhẹ, có rất ít lỗi do ít bị phụ thuộc cũng như dễ dàng phát hiện và xử lý các lỗi bảo mật.

3.2. Thuật toán CNN

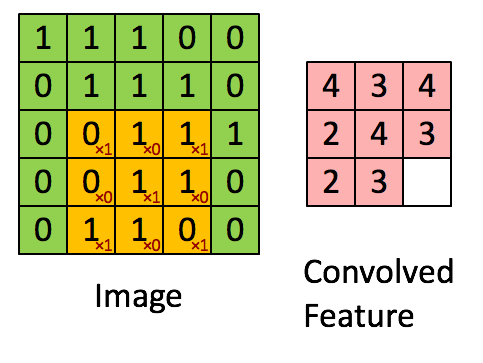
**Convolutional Neural Network là gì**

Convolutional Neural Network (CNNs – Mạng nơ-ron tích chập) là một trong những mô hình Deep Learning tiên tiến. Nó giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay.

CNN được sử dụng nhiều trong các bài toán nhận dạng các object trong ảnh. Để tìm hiểu tại sao thuật toán này được sử dụng rộng rãi cho việc nhận dạng (detection), chúng ta hãy cùng tìm hiểu về thuật toán này.

**Convolutional là gì?**

Là một cửa sổ trượt (Sliding Windows) trên một ma trận như mô tả hình dưới:

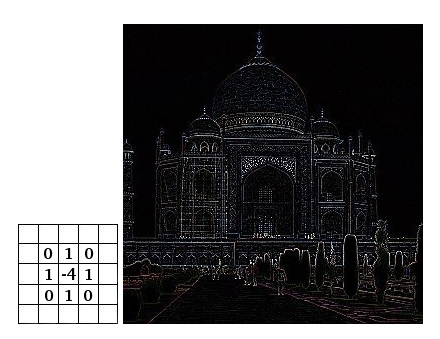


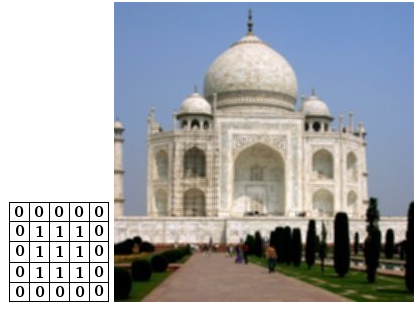
Các convolutional layer có các parameter(kernel) đã được học để tự điều chỉnh lấy ra những thông tin chính xác nhất mà không cần chọn các feature.

Trong hình ảnh ví dụ trên, ma trận bên trái là một hình ảnh trắng đen được số hóa. Ma trận có kích thước 5×5 và mỗi điểm ảnh có giá trị 1 hoặc 0 là giao điểm của dòng và cột.

Convolution hay tích chập là nhân từng phần tử trong ma trận 3. Sliding Window hay còn gọi là kernel, filter hoặc feature detect là một ma trận có kích thước nhỏ như trong ví dụ trên là 3×3.

Convolution hay tích chập là nhân từng phần tử bên trong ma trận 3×3 với ma trận bên trái. Kết quả được một ma trận gọi là Convoled feature được sinh ra từ việc nhận ma trận Filter với ma trận ảnh 5×5 bên trái.





**Cấu trúc của mạng CNN**

Mạng CNN là một tập hợp các lớp Convolution chồng lên nhau và sử dụng các hàm nonlinear activation như ReLU và tanh để kích hoạt các trọng số trong các node. Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo.

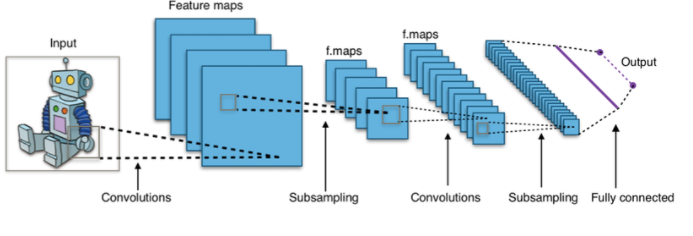
Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo. Trong mô hình mạng truyền ngược (feedforward neural network) thì mỗi neural đầu vào (input node) cho mỗi neural đầu ra trong các lớp tiếp theo.

Mô hình này gọi là mạng kết nối đầy đủ (fully connected layer) hay mạng toàn vẹn (affine layer). Còn trong mô hình CNNs thì ngược lại. Các layer liên kết được với nhau thông qua cơ chế convolution.

Layer tiếp theo là kết quả convolution từ layer trước đó, nhờ vậy mà ta có được các kết nối cục bộ. Như vậy mỗi neuron ở lớp kế tiếp sinh ra từ kết quả của filter áp đặt lên một vùng ảnh cục bộ của neuron trước đó.

Mỗi một lớp được sử dụng các filter khác nhau thông thường có hàng trăm hàng nghìn filter như vậy và kết hợp kết quả của chúng lại. Ngoài ra có một số layer khác như pooling/subsampling layer dùng để chắt lọc lại các thông tin hữu ích hơn (loại bỏ các thông tin nhiễu).

Trong quá trình huấn luyện mạng (traning) CNN tự động học các giá trị qua các lớp filter dựa vào cách thức mà bạn thực hiện. Ví dụ trong tác vụ phân lớp ảnh, CNNs sẽ cố gắng tìm ra thông số tối ưu cho các filter tương ứng theo thứ tự raw pixel > edges > shapes > facial > high-level features. Layer cuối cùng được dùng để phân lớp ảnh.

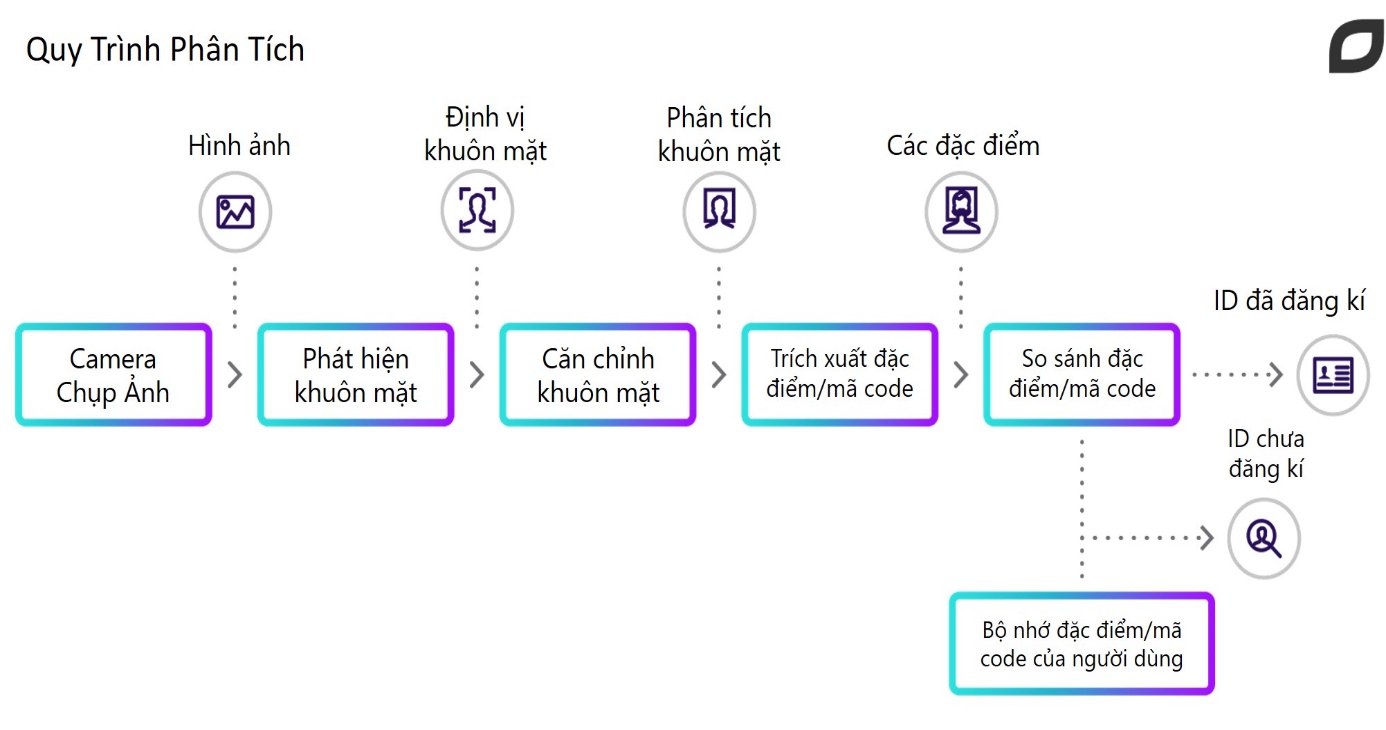


Trong mô hình CNN có 2 khía cạnh cần quan tâm là tính bất biến (Location Invariance) và tính kết hợp (Compositionality). Với cùng một đối tượng, nếu đối tượng này được chiếu theo các gốc độ khác nhau (translation, rotation, scaling) thì độ chính xác của thuật toán sẽ bị ảnh hưởng đáng kể.

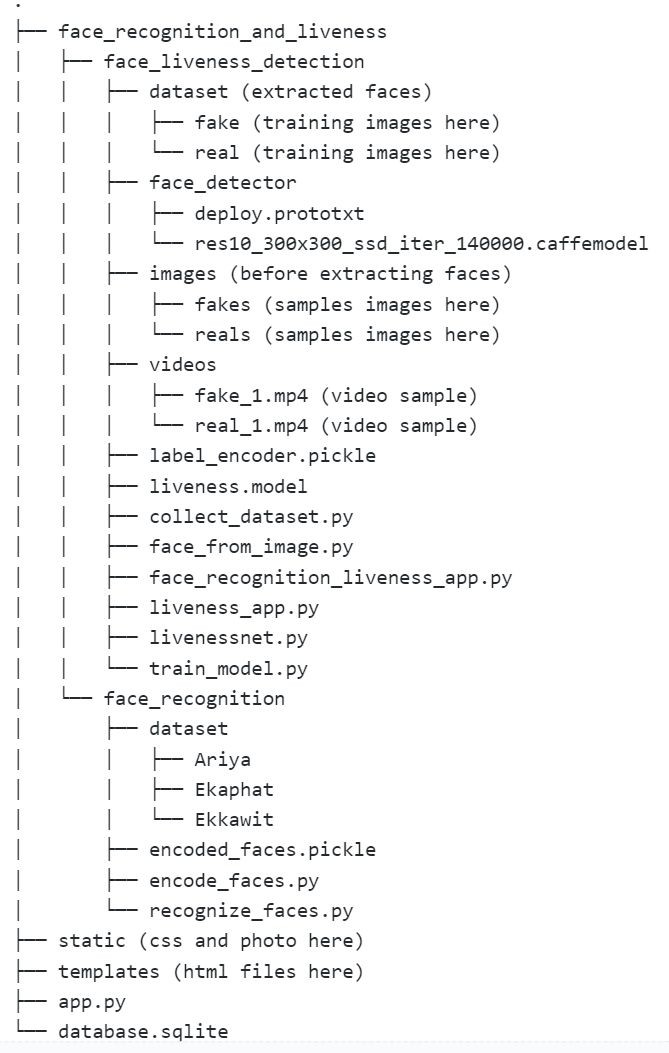
Pooling layer sẽ cho bạn tính bất biến đối với phép dịch chuyển (translation), phép quay (rotation) và phép co giãn (scaling). Tính kết hợp cục bộ cho ta các cấp độ biểu diễn thông tin từ mức độ thấp đến mức độ cao và trừu tượng hơn thông qua convolution từ các filter.

Đó là lý do tại sao CNNs cho ra mô hình với độ chính xác rất cao. Cũng giống như cách con người nhận biết các vật thể trong tự nhiên.

## 3.3.Các bước tiến hành



## 3.4.Mô tả dự án

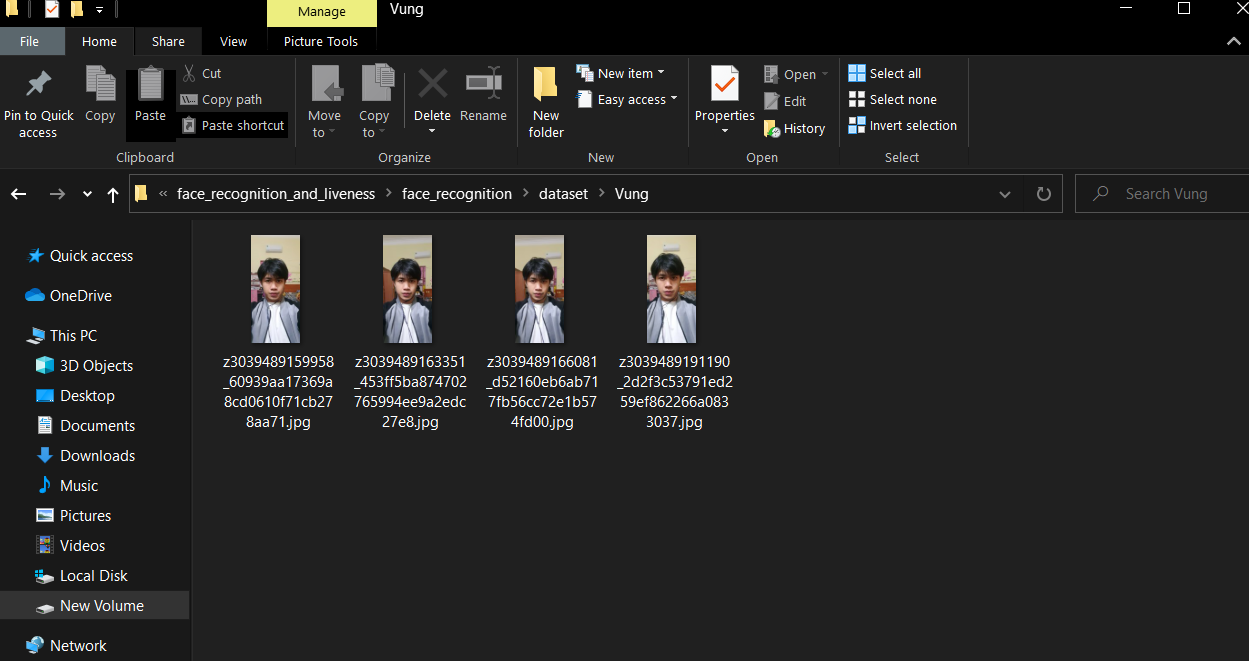


# CHƯƠNG 4: Cài đặt và kiểm thử

## 4.1.Triển khai bài toán

## 4.1.1.Chuẩn bị dữ liệu

Chúng ta chuẩn bị đầu vào một số hình ảnh khuôn mặt



Dữ liệu đàu vào bao gồm 4 ảnh cá nhân tự chụp

## 4.1.2.Train mô hình

Sau khi có dataset thì chúng ta train dữ liệu theo mô hình YOLOV5.

Chúng ta có các bước train data theo mô hình CNN:

Tạo 1 thư mục cho 1 người và đặt tên theo tên của người đó trong face\_recognition / dataset (bạn có thể xem thư mục này trong repo này chẳng hạn)

Thu thập các hình ảnh hiển thị đầy đủ khuôn mặt (1 khuôn mặt trên 1 hình ảnh trên 1 người). Vì chúng tôi đang sử dụng kỹ thuật học chụp 1 lần , nên chỉ thu thập tối đa 10 hình ảnh cho mỗi người là đủ.

Chạy encode\_faces.py Bây giờ bạn sẽ nhận được tệp khuôn mặt được mã hóa kết thúc bằng .pickle trong đường dẫn bạn chỉ định (nếu bạn làm theo mã ở trên, bạn sẽ thấy nó trong cùng một thư mục với tệp này)

Chạy recognize\_faces.py xem liệu nó có hoạt động tốt hay không.

Thu thập video của chính bạn / người khác trong nhiều điều kiện ánh sáng (cách dễ nhất để làm điều này là quay phim chính bạn / người khác đang đi quanh nhà của bạn / những người khác) và lưu vào thư mục face\_liveness\_dection / videos . Độ dài của video tùy thuộc vào bạn. Bạn không cần phải đặt tên cho nó bằng từ 'thật' hoặc 'giả'. Đó chỉ là quy ước mà chúng tôi thấy hữu ích khi gọi từ các mã khác. Hãy xem thư mục đó, chúng tôi đã bỏ một số video ví dụ ở đó.

Sử dụng những video đã ghi đó và phát nó trên điện thoại của bạn. Sau đó, giữ điện thoại của bạn và hướng màn hình điện thoại (đang chạy các video đã quay đó) vào webcam và quay màn hình PC / laptop của bạn. Bằng cách làm này, bạn đang tạo tập dữ liệu về ai đó giả mạo người trong video / giả mạo là người trong video. Cố gắng đảm bảo video giả mạo mới này có cùng độ dài (hoặc gần) với video gốc vì chúng ta cần tránh tập dữ liệu không cân bằng . Hãy xem thư mục đó, chúng tôi đã bỏ một số video ví dụ ở đó.

Chạy collect\_dataset.py. Đảm bảo rằng bạn lưu đầu ra vào đúng thư mục (phải nằm trong datasetthư mục và trong thư mục nhãn bên phải fakehoặc real). Bây giờ bạn phải xem rất nhiều hình ảnh từ video của mình trong thư mục đầu ra.

(Tùy chọn, nhưng nên làm để cải thiện hiệu suất của mô hình) Chụp ảnh bạn / người khác từ giấy, ảnh, thẻ, v.v. và lưu vào face\_liveness\_detection / images / fakes . Hãy xem thư mục đó, chúng tôi đã bỏ một số video ví dụ ở đó.

Nếu bạn làm bước 9, vui lòng thực hiện bước này. Nếu không, bạn có thể bỏ qua bước này. Chụp thêm ảnh khuôn mặt của bạn / người khác với cùng lượng ảnh giả mà bạn đã chụp ở bước 8 và lưu vào face\_liveness\_detection / images / reals . Một lần nữa, bằng cách làm này, chúng ta có thể tránh được tập dữ liệu không cân bằng . Hãy xem thư mục đó, chúng tôi đã bỏ một số video ví dụ ở đó.

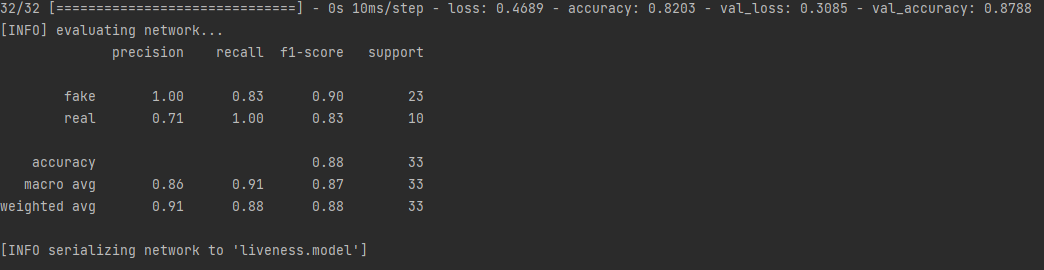
(Bỏ qua bước này nếu bạn không thực hiện bước 9) Chạy face\_from\_image.py cho mọi hình ảnh của bạn trong thư mục hình ảnh. Đảm bảo rằng bạn lưu đầu ra vào đúng thư mục (phải nằm trong datasetthư mục và trong thư mục nhãn bên phải fakehoặc real). Lưu ý: Giống như chúng ta đã thảo luận trong phần giải thích tệp, bạn phải chạy mã này 1 hình ảnh tại một thời điểm. Nếu bạn có nhiều hình ảnh, vui lòng điều chỉnh mã. Vì vậy, bạn chỉ có thể chạy một lần cho mỗi hình ảnh của mình. (Nhưng hãy đảm bảo lưu kết quả đầu ra vào đúng thư mục)

Chạy train\_model.py. Bây giờ, chúng ta sẽ có .model, tệp bộ mã hóa nhãn kết thúc bằng .pickle và hình ảnh trong thư mục đầu ra mà bạn chỉ định. Nếu bạn làm theo mã chính xác ở trên trong phần giải thích tệp, bạn sẽ thấy liveness.model, label\_encoder.pickle và plot.png trong thư mục chính xác này (như trong repo này).

Chạy liveness\_app.py và xem liệu nó có hoạt động tốt hay không. Nếu mô hình luôn phân loại sai, hãy quay lại và xem liệu bạn có lưu hình ảnh đầu ra (thật / giả) vào đúng thư mục hay không. Nếu bạn chắc chắn rằng bạn lưu mọi thứ vào đúng vị trí, hãy thu thập nhiều dữ liệu hơn hoặc dữ liệu chất lượng tốt hơn. Đây là quá trình lặp đi lặp lại phổ biến của mô hình đào tạo, đừng cảm thấy tồi tệ nếu bạn gặp vấn đề này.

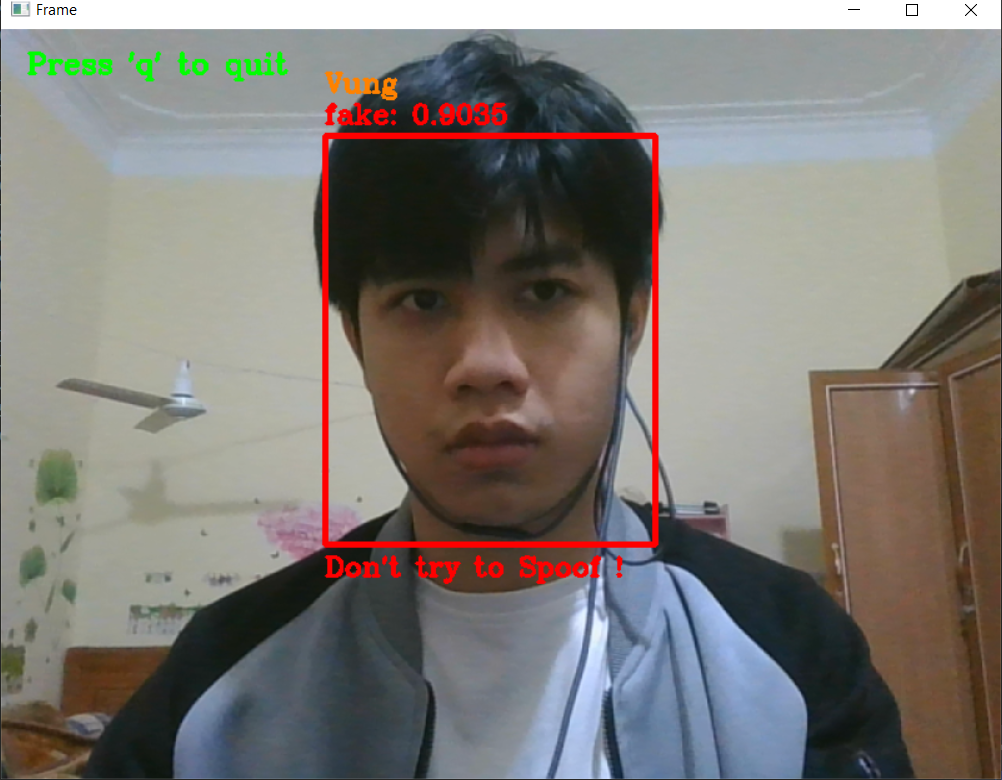
Chạy face\_recognition\_liveness\_app.py và xem liệu nó có hoạt động tốt hay không. Cửa sổ sẽ hiển thị một lúc với hộp giới hạn có tên của bạn ở trên cùng và xác suất là thật hay giả. Nếu công việc mã của bạn đúng cách, cửa sổ này sẽ tự động bị đóng sau khi mô hình có thể phát hiện bạn và bạn là có thật cho 10 khung hình quả . Bằng cách chỉ định 10 khung hệ quả này, nó có thể đảm bảo rằng người trên màn hình thực sự là người đang đăng nhập và có thật, chứ không chỉ do ngẫu nhiên mà mô hình phân loại sai cho một số khung. Nó cũng cho phép một số chỗ để mô hình phân loại sai. Bạn có thể điều chỉnh con số này trong biến dòng 176 'serial\_count'trong mã, nếu bạn muốn nó dài hơn hoặc ngắn hơn. Và ngay sau khi cửa sổ đóng lại, chúng ta sẽ thấy tên và nhãn (thật) của bạn trên dòng lệnh / thiết bị đầu cuối.

Sau khi train ta có kết quả như sau:



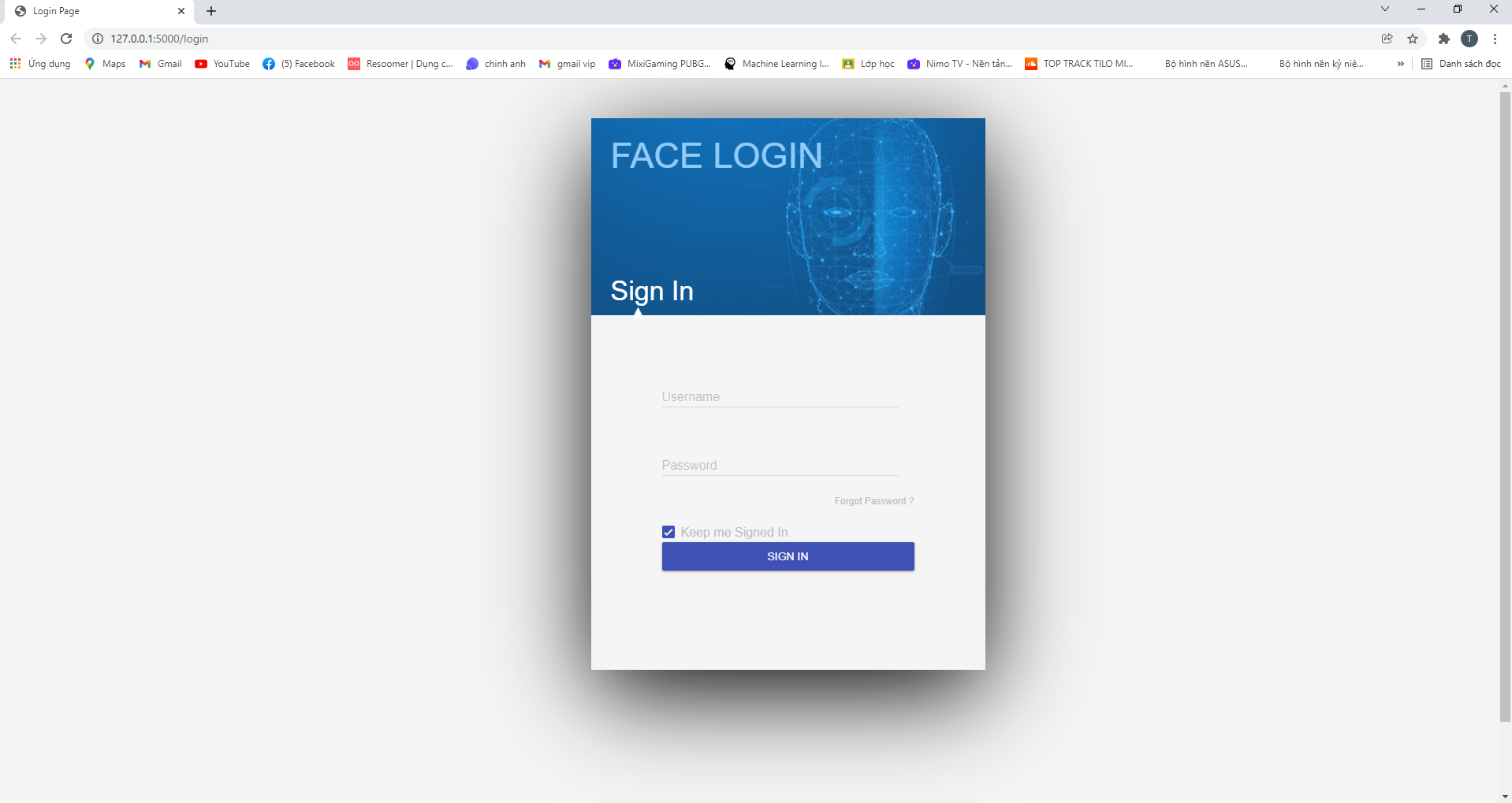
## 4.1.3. kết quả dự đoán

Kết quả thu được là dự đoán người đang đăng nhập:

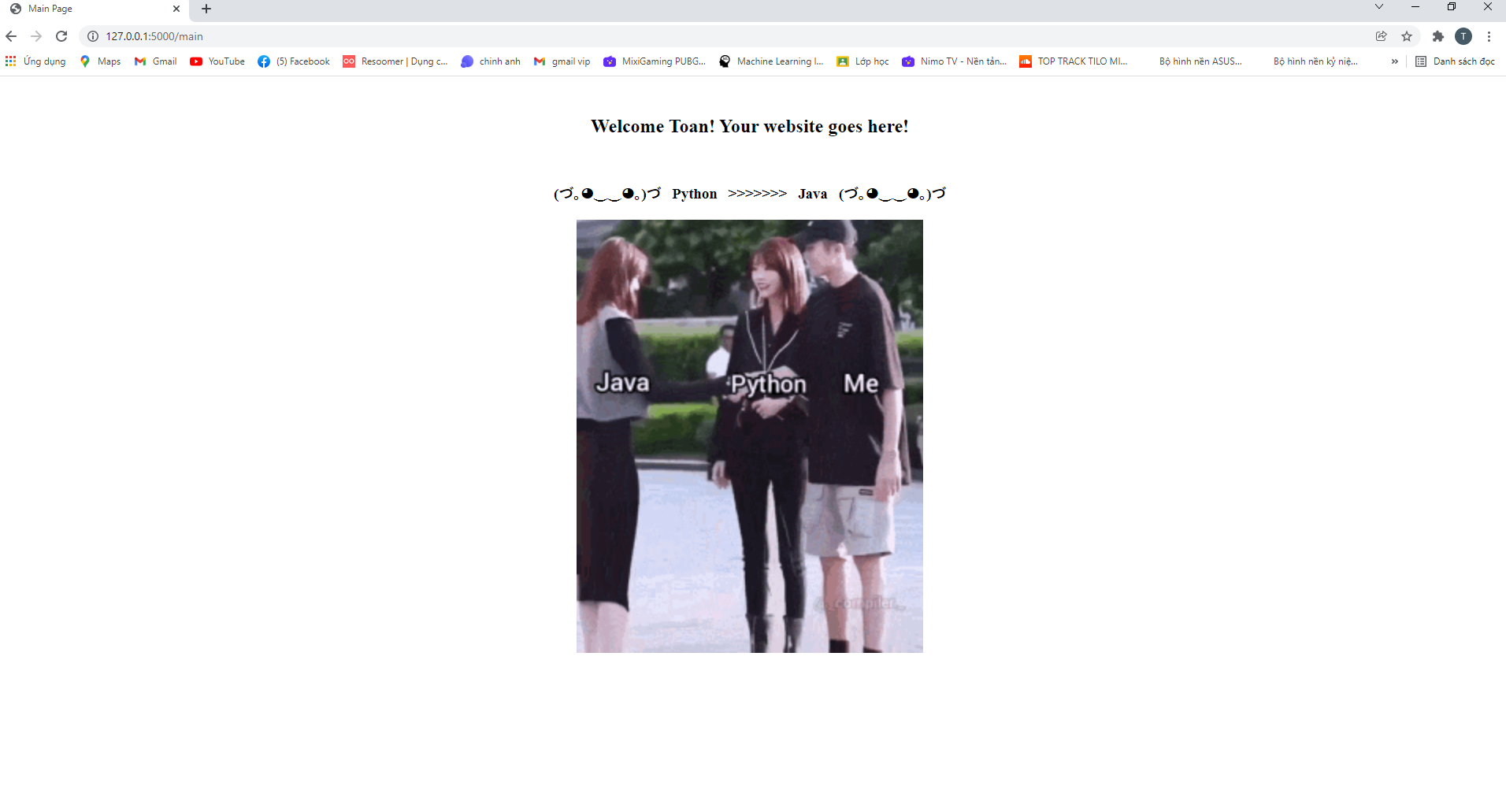
****

## 4.1.4.trang web face login

Giao diện trang login**:**



Giao diện tranng index:



# KẾT LUẬN

Trong quá trình thực hiện bài tập lớn của môn nhập môn học máy, chúng em đã được biết thêm về các chương trình ứng dụng, nắm rõ về các phần về học máy. Từ đó chúng em cố gắng áp dụng các kiến thức đã học vào làm bài tập để hoàn thiện sản phẩm của mình. Trong bài tập lớn của chúng em sử dụng thuật toán CNN đê nhận dạng khuôn mặt . Vì trong thời gian ngắn nên bài tập lớn của chúng em vẫn còn nhiều thiếu sót. Chúng em rất hy vọng có sự góp ý của thầy cô, để bài tập lớn của chúng em được hoàn thiện hơn nữa ạ

# DANH MỤC TÀI LIỆU THAM KHẢO

1. [ML] Hồi quy tuyến tính : https://dominhhai.github.io/vi/2017/12/ml-linear-regression/
2. Machine Learning cơ bản: https://machinelearningcoban.com/2016/12/28/linearregression/
3. Phân tích hồi quy tuyến tính đa biến : <https://bit.ly/30xi3Vb>
4. 1. Schneider A, Hommel G, Blettner M. Linear regression analysis: part 14 of a series on evaluation of scientific publications. Dtsch Arztebl Int. 2010 Nov;107(44):776-82.
5. Tripepi G, Jager KJ, Stel VS, Dekker FW, Zoccali C. How to Deal with Continuous and Dichotomic Outcomes in Epidemiological Research: Linear and Logistic Regression Analyses. Nephron Clin Pract. 2011 Feb 23;118(4):c399- c406.
6. Hoàng Trọng và Chu Nguyễn Mộng Ngọc, Phân tích dữ liệu nghiên cứu với SPSS. Nhà xuất bản thống kê năm 2005