TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

———— * ————

BÁO CÁO ĐỒ ÁN II

Project II

DEEP LEARNING:
NHẬN DẠNG KHUÔN MẶT

Giáo viên hướng dẫn : **TS. Đỗ Phan Thuận**

Nhóm sinh viên thực hiện :

STT	Họ và tên	MSSV
1	Trần Trung Hiếu	20141567
2	Lê Trần Bảo Cương	20140542
3	Vũ Trọng Hiệu	20141683

Hà Nội, tháng 6 năm 2017

MỤC LỤC

LỜI NÓI ĐẦU

Trong những năm gần đây, trí tuệ nhân tạo (AI) đang thực sự bùng nổ nhờ các thành tựu đột phá trong lĩnh vực học máy và khoa học dữ liệu. Hàng loạt sản phẩm thông minh áp dụng trí tuệ nhân tạo đã giúp cải thiện năng suất lao động, nâng cao chất lượng cuộc sống con người, tiết kiệm chi phí nhân lực cho doanh nghiệp và xã hội... Nhiều ứng dụng trí tuệ nhân tạo trong các lĩnh vực như thị giác máy tính, nhận dạng giọng nói hay xử lý ngôn ngữ tự nhiên đã đạt đến độ chính xác ngang bằng hoặc thâm chí vượt con người trong một số trường hợp. Trong các bài toán ấy, nhóm chúng em nhận thấy bài toán "Nhận dạng khuôn mặt người" có rất nhiều điều mới mẻ, phù hợp áp dụng các kiến thức về học máy/ học sâu mà nhóm đang tìm hiểu. Vì vậy, nhóm quyết định chọn đề tài này cho Đồ án 2.

Cụ thể, chúng em tìm hiểu các vấn đề phát hiện khuôn mặt trong một bức ảnh, video; giải quyết bài toán xác thực khuôn mặt (face verification) để xác minh hai bức ảnh cho trước có phải thuộc cùng một người hay không.

Ngoài ra, nhóm quyết định tìm hiểu các kiến thức về Javascript gồm cả front-end và back-end để có thể xây dựng hệ thống trên nền tảng Web với NodeJS.

Để hoàn thành được Đồ án 2 này, nhóm em xin được gửi lời cảm ơn chân thành đến TS. Đỗ Phan Thuận – giảng viên Viện CNTT&TT, Đại học Bách khoa Hà Nội – đã hướng dẫn, góp ý tận tình, cung cấp tài liệu tham khảo để nhóm có thể hoàn thành đề tài này.

Hà Nội, tháng 6 năm 2017

PHẦN 1: PHÁT HIỆN VÀ CĂN CHỈNH KHUÔN MẶT

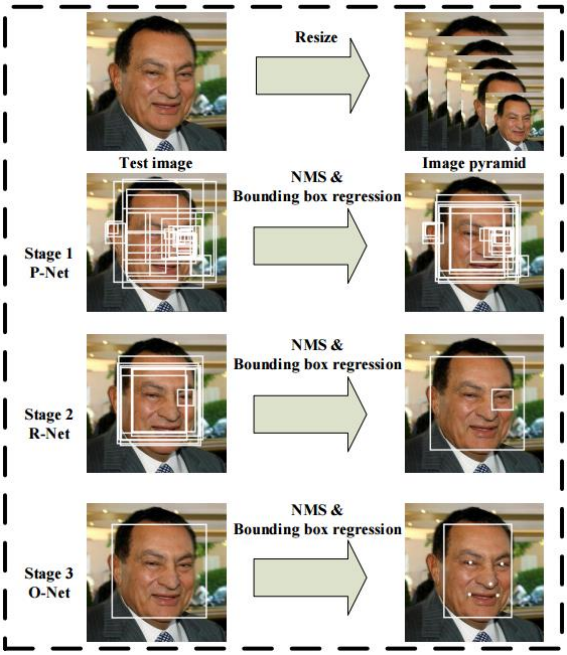
1. Tổng quan và so sánh

Nhận dạng và căn chỉnh khuôn mặt là bài toán đã được nghiên cứu từ rất lâu bằng nhiều thuật toán học máy khác nhau. Các thuật toán nhận dạng thường gặp phải thách thức do các góc độ hay điệu bộ khác nhau của khuôn mặt, vấn đề khuôn mặt bị che mất một phần, bóng do chiếu sáng từ một bên… Các thuật toán trong thư viện xử lí ảnh trước đây như OpenCV, dlib thường nhận dạng sai hoặc không nhận dạng được trong các trường hợp phức tạp trên, phần nào ảnh hưởng tới hiệu quả của các xử lí sau này. Các nghiên cứu gần đây cho thấy hướng tiếp cận deep learning có thể đạt được hiệu quả ấn tượng khi áp dụng vào hai bài toán trên. Ở đây, nhóm em đã tìm hiểu một phương pháp mới sử dụng deep learning để khai thác mối tương quan vốn có giữa nhận dạng và căn chỉnh để tăng hiệu quả cả về thời gian lẫn độ chính xác. Đó là mạng tích chập Multi-task CNNs.

2. Multi-task Convolution Neural Networks

2.1. Cấu trúc mạng

Mạng này bao gồm 3 tầng CNNs. Ở tầng đầu tiên, mạng đề xuất nhanh các "cửa sổ" làm ứng cử viên cho quá trình tìm ra khuôn mặt thực sự thông qua một mạng CNN mỏng, tầng này được gọi là P-Net. Tiếp theo, mạng này lọc tiếp các "cửa sổ" bằng cách bỏ qua lượng lớn các cửa sổ không phải khuôn mặt nhờ một mạng CNN phức tạp hơn, là R-Net. Cuối cùng, nó sử dụng một mạng CNN mạnh mẽ nhất để lọc kết quả một lần nữa và xuất ra kết quả kèm theo 5 điểm landmark trên khuôn mặt gồm 2 mắt, đỉnh mũi và 2 mép của miệng.



Hình 1.

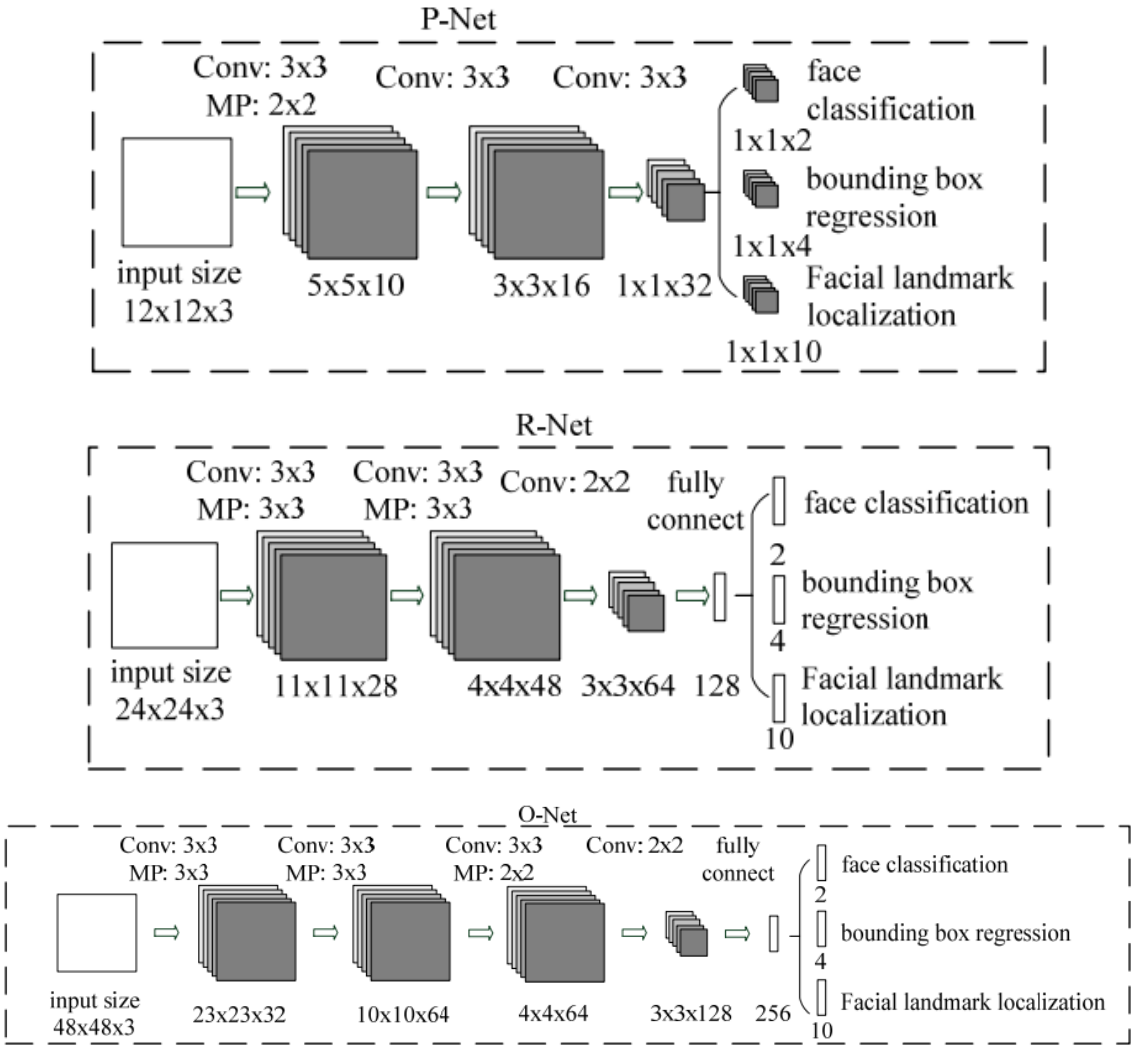
2.2. Tổng quan

Tổng quan của mô hình được minh họa bởi Hình 1. Nhận vào một bức ảnh, nó sẽ resize sang nhiều kích cỡ tỉ lệ khác nhau và tạo thành "tháp" ảnh, là đầu vào của mạng 3 tầng sau:

- Tầng 1 – P-Net: Đây là tầng mạng kết nối đầy đủ (fully connected), để tìm ra các "cửa sổ ứng cử viên" và các vector hồi quy tương ứng. Sau đó các ứng cử viên được hiệu chỉnh dựa trên vector hồi quy để dự đoán. Cuối cùng, mạng sử dụng non-maximum suppression (NMS) để hợp nhất các cửa sổ chồng chéo nhau.
- Tầng 2 – R-Net: Tất cả các ứng cử viên được đưa qua mạng CNN thứ 2 để loại bỏ phần lớn các cửa sổ sai, tiến hành hiệu chuẩn với hồi quy.
- Tầng 3 – O-Net: Tầng này giống với tầng 2, nhưng ở tầng này mô hình cố gắng tìm ra vị trí của khuôn mặt trong cửa sổ. Cụ thể, mạng sẽ xuất ra 5 điểm lardmark của một khuôn mặt.

2.3. Cấu trúc CNN

Mạng CNN sử dụng các filter 3x3 thay vì 5x5 và Max pooling sau mỗi filter để giảm chi phí tính toán và tăng độ sâu để đạt kết quả cao hơn. Mạng sử dụng PReLU làm hàm activation function sau mỗi tầng conv và tầng fully connection.



2.4. Huấn luyện

Mạng phân chia thành 3 tác vụ để huấn luyện mạng CNN: face/non-face classification, bounding box regression và facial landmark localization.

1. Face classification: Quá trình học được xây dựng như bài toán phân loại 2 lớp. Với mỗi đầu vào x_i, mạng sử dụng hàm chi phí cross-entropy:

$$L_i^{det} = -(y_i^{det} \log(p_i) + (1 - y_i^{det})(1 - \log(p_i)))$$

Trong đó p_i là xác suất mô hình đoán nhận x_i là một khuôn mặt, y_i^{det} là nhãn thực tế.

2. Bounding box regression: Với mỗi cửa sổ ứng cử viên, mạng dự đoán offset giữa nó và nhãn đúng gần nhất. Quá trình học được xây dựng như bài toán hồi quy, sử dụng hàm chi phí Euclid cho mỗi đầu vào x_i:

$$L_i^{box} = \left\| \hat{y}_i^{box} - y_i^{box} \right\|_2^2$$

Trong đó $\hat{y}_i^{box}$ là lớp mà mô hình đoán nhận và y_i^{box} là nhãn thực tế.

3. Facial landmark localization: Giống như bounding box regression, facial landmark cũng được xây dựng như bài toán hồi quy và sử dụng hàm chi phí Euclidean:

$$L_i^{landmark} = \left\| \hat{y}_i^{landmark} - y_i^{landmark} \right\|_2^2$$

Có 5 điểm facial landmark, bao gồm mắt trái, mắt phải, mũi, mép miệng trái và mép miệng phải.

4. Multi-source training: Mạng thực hiện nhiều nhiệm vụ khác nhau trong mỗi tầng CNN, có nhiều cách thức huấn luyện khác nhau với các hàm chi phí cụ thể. Nên khi tổng hợp lại toàn bộ mô hình thì không cần hàm chi phí nữa. Cụ thể, quá trình huấn luyện được xây dựng:

$$\min \sum_{i=1}^{N} \sum_{j \in \{det, box, landmark\}} \alpha_j \beta_i^j L_i^j$$

Trong đó, N là số lượng ví dụ tập trainning, a_j biểu thị tầm quan trọng của tách vụ. Mạng sử dụng ($a_{det} = 1$, $a_{box} = 0.5$, $a_{landmark} = 0.5$) in P-Net và R-Net trong khi ($a_{det} = 1$, $a_{box} = 0.5$, $a_{landmark} = 1$) trong O-Net.
Mô hình sử dụng Stochastic gradient descent để huấn luyện CNNs.

PHẦN 2: ÁNH XẠ KHÔNG GIAN KHUÔN MẶT NGƯỜI SANG KHÔNG GIAN VECTOR

1. Tổng quan

Sau khi đã giải quyết được bài toán nhận dạng và căn chỉnh khuôn mặt, xây dựng được mô hình đủ tốt với độ chính xác cao. Nhóm bắt đầu tìm hiểu giải pháp cho các bài toán liên quan nhận dạng khuôn mặt. Yêu cầu đặt ra là làm thế nào để có thể tạo ra một mô hình đồng nhất cho phép giải quyết các loại bài toán liên quan tới nhận dạng khuôn mặt. Sau khi tìm hiểu một số phương pháp, nhóm quyết định áp dụng kỹ thuật học sâu deep learning cho phép ánh xạ không gian khuôn mặt người vào một không gian vector với số chiều định trước (cụ thể là 128 chiều). Mô hình cho phép huấn luyện tham số của phép ánh xạ nhằm đảm bảo tính chất sau: các bức ảnh của cùng một người sẽ được ánh xạ thành những điểm gần nhau trong không gian mới; trong khi đó các bức ảnh thuộc những người khác nhau sẽ nằm phân tán cách xa nhau một khoảng cách nhất định. Sử dụng phương pháp này chúng ta có thể dễ dàng giải quyết các bài toán khác nhau liên quan tới nhận dạng khuôn mặt một cách dễ dàng bằng cách xử lý dữ liệu trực tiếp trên không gian ánh xạ mới.

2. Mô hình mạng FaceNet

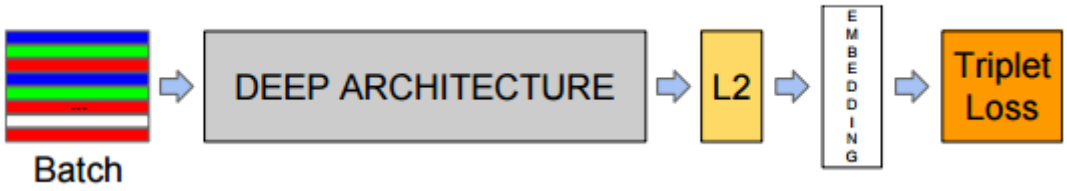
FaceNet sử dụng một mạng tích chập CNNs sâu cho phần ánh xạ. Ban đầu, mạng được nghiên cứu trên hai kiến trúc chính: mạng nhiều tầng của Zeiler & Fergus và mạng Inception. Gần đây, khi mạng ResNet ra đời cùng với các thành tựu vượt trội state-of-the-art thì FaceNet đã được xây dựng bằng mô hình kết hợp giữa Inception với ResNet, có tên gọi Inception-ResNet-v1. Nhóm quyết định tìm hiểu và xây dựng dựa theo mô hình này.

2.1. Cấu trúc mạng

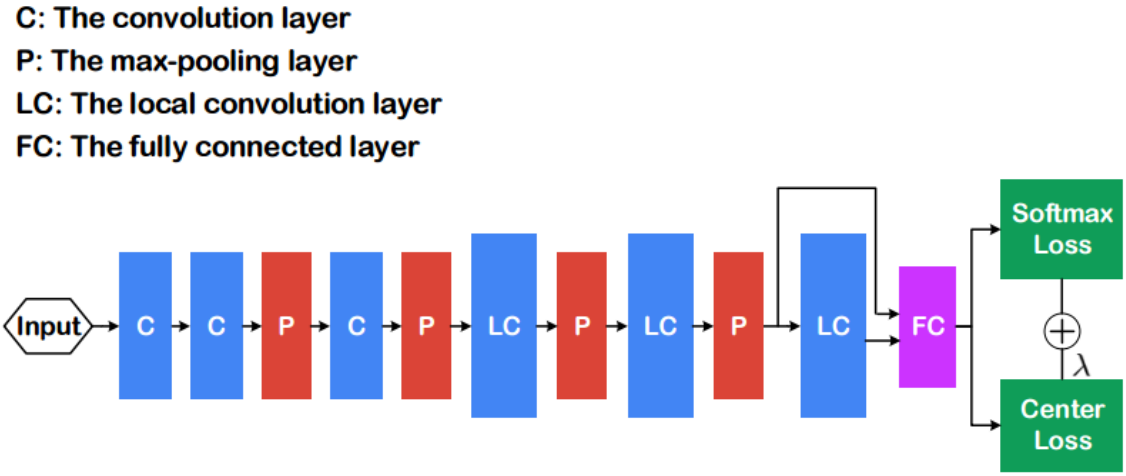
Mạng gồm các thành phần chính:

- Convolution layer
- Max-pooling layer
- Local convolution layer
- Fully connected layer

Sau khi xây dựng được mạng cho phần ánh xạ Embedding, mô hình cần được xây dựng hàm chi phí – loss function thích hợp để có thể đạt được những yêu cầu đề ra. Đó là các ảnh của cùng một người phải ánh xạ về các điểm gần nhau, ngược lại, những ảnh của 2 người khác nhau phải đảm bảo khoảng cách nhất định. FaceNet ban đầu sử dụng Triplet Loss để đánh giá. Gần đây một nhóm tác giả đã đưa ra hàm Center Loss với kết quả tốt hơn.



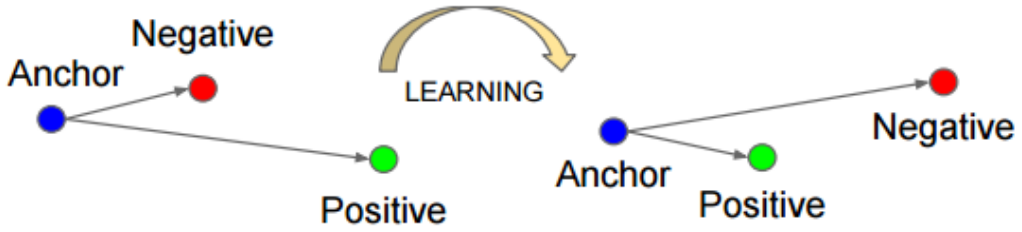
Cấu trúc mạng FaceNet ban đầu.



FaceNet sử dụng Center Loss

2.2. Triplet Loss

Hàm ánh xạ được biểu diễn bởi hàm $f(x) \in \mathbb{R}^d$. Nó ánh xạ một bức ảnh x thành một vector d-chiều trong không gian Euclid. Ở đây, FaceNet muốn đảm bảo rằng một bức ảnh x_i^a (anchor) của một người cụ thể là gần với tất cả các ảnh khác của người đó x_i^p (positive) hơn so với những ảnh của bất kì ai khác. Mục đích của quá trình training là đảm bảo điều này.



Tóm lại, cần đảm bảo

$$\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2 ,$$

$$\forall (f(x_i^a), f(x_i^p), f(x_i^n)) \in \mathcal{T} .$$

Trong đó, α là ngưỡng khoảng cách để phân định giữa một cặp positive và negative. T là tập tất cả các bộ ba có thể có trong training set, số lượng là N.

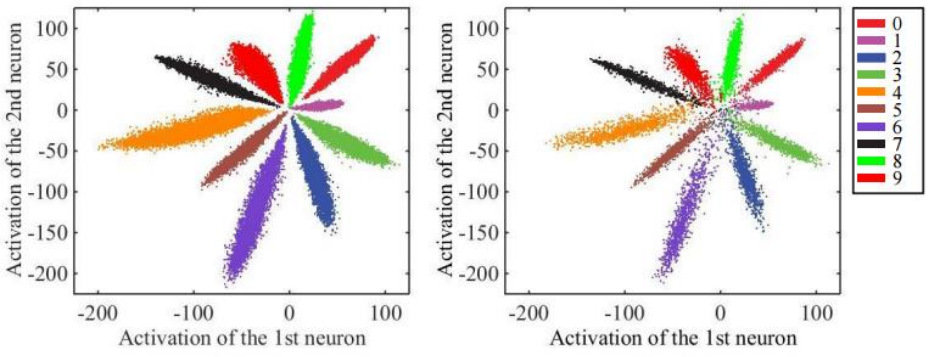
Vậy mục đích cần đạt là cực tiểu hóa nên L =

$$\sum_i^N \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha\right]_+$$

Nếu ta tạo ra tất cả các bộ ba (triplet) có thể thì sẽ xuất hiện các bộ ba thỏa mãn từ trước. Tức là chúng không đóng góp gì cho việc huấn luyện mà lại làm chậm quá trình hội tụ, vì vẫn được lan truyền qua mạng. Việc lựa chọn đúng các bộ ba là điều rất cần thiết, góp phần xây dựng mô hình nhanh chóng hội tụ.

2.3. Center Loss

Mô hình sử dụng Triplet Loss có những kết quả nhất định, tuy nhiên kết quả vẫn chưa cao. Bằng việc biểu diễn lại các vector sau khi ánh xạ ảnh mặt của những người khác nhau thành các vector 2 chiều để quan sát tổng thể. Người ta nhận thấy rằng tuy các tâm của từng nhóm ảnh đã xa nhau nhưng vẫn tồn tại những điểm rất gần nhau (Hình minh họa).



(a) (b)

Từ đó, người ta nhận thấy không chỉ khoảng cách của các tâm cần xa nhau, mà khoảng cách giữa các điểm trong cùng một lớp cũng cần gần lại. Tức là cố gắng đảm bảo các điểm của một lớp tụ lại gần nhau hơn.

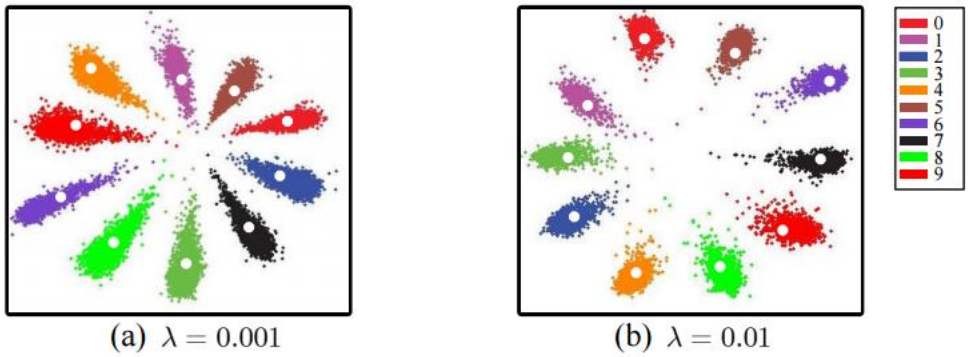
Để làm được điều đó, FaceNet định nghĩa 2 hàm chi phí khác nhau: Center loss và Softmax Loss.

- Center Loss (L_C) tính khoảng cách giữa các điểm của cùng một lớp sau khi ánh xạ. Center Loss nhỏ đảm bảo các điểm tụ về gần tâm hơn.
- Softmax Loss (L_S) tính khoảng cách giữa các lớp khác nhau. Cực tiểu chi phí này giúp đảm bảo yêu cầu cơ bản của FaceNet.

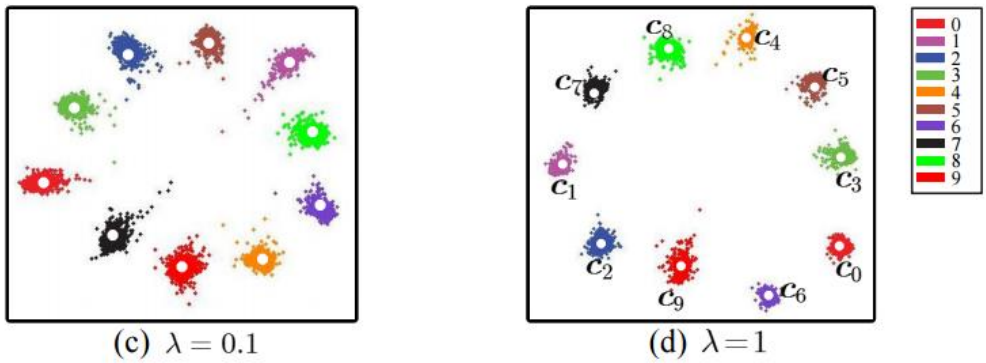
$$\mathcal{L}_C = \frac{1}{2}\sum_{i=1}^{m}\|\boldsymbol{x}_i - \boldsymbol{c}_{y_i}\|_2^2$$

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_S + \lambda\mathcal{L}_C \\ &= -\sum_{i=1}^{m}\log\frac{e^{W_{y_i}^T\boldsymbol{x}_i+b_{y_i}}}{\sum_{j=1}^{n}e^{W_j^T\boldsymbol{x}_i+b_j}} + \frac{\lambda}{2}\sum_{i=1}^{m}\|\boldsymbol{x}_i - \boldsymbol{c}_{y_i}\|_2^2\end{aligned}$$

Hệ số λ giúp ta điều chỉnh độ ưu tiên giữa 2 loại chi phí trên, λ càng lớn thì chi phí Center loss càng được đánh giá cao. Việc lựa chọn λ cần thông qua thực nghiệp để phụ hợp với mô hình.



(a) $\lambda = 0.001$ (b) $\lambda = 0.01$



(c) $\lambda = 0.1$ (d) $\lambda = 1$

Ở đây, ta thấy với λ trong khoảng 0.1 đến 1 thì kết quả là tốt nhất.

PHẦN 3: WEBSITE

I. Back-end

Đối với phần server, nhóm sử dụng nền tảng NodeJS, cụ thể là sử dụng framework ExpressJS cho việc xử lý các request. Với phần nghiệp vụ của website là nhận diện mặt chủ yếu làm việc với Python, do đó để gọi được code python nhóm sử dụng thư viện Python-shell. Với phần cơ sở dữ liệu, nhóm sử dụng thư viện MongooseJS để giao tiếp với MongoDB. Sau đây sẽ là phần giới thiệu chi tiết về các công nghệ sử dụng.

1. Nodejs

- Node.js là gì ?

NodeJS là một nền tảng Server-side mã nguồn mở, mỗi trường thực thi chạy được trên nhiều nền tảng như Windows, Linux, OS X … được xây dựng dựa trên Javascript Engine (V8 Engine). Node.js được phát triển bởi Ryan Dahl năm 2009. Định nghĩa NodeJs bởi tài liệu chính thức như sau:

> Node.js là một nền tảng dựa vào Chrome Javascript runtime để xây dựng các ứng dụng mạng nhanh và có khả năng mở rộng. Node.js sử dụng các phần phát sinh các sự kiện (event-driven), mô hình non-blocking I/O để tạo ra các ứng dụng nhẹ và hiệu quả cho các ứng dụng dữ liệu lớn thời gian thực chạy trên các thiết bị phân tán.

Node.js cũng cung cấp cho chúng ta một thư viện rất phong phú các module Javascript, có thể đơn giản hóa sự phát triển của các ứng dụng web sử dụng Node.js tới một quy mô lớn.

```
Node.js = Runtime Environment + JavaScript Library
```

- Đặc trưng của Node.js

Dưới đây là vài đặc điểm quan trọng biến Node.js trở thành sự lựa chọn hàng đầu của các nhà thiết kế:

- **Không đồng bộ và Hướng sự kiện (Asynchronous and Event Driven)** : Tất các các APIs của thư viện Node.js đều không đồng bộ, nghĩa là non-blocking. Về bản chất một Node.js server không bao giờ đợi một API trả về dự liệu. Server chuyển sang một API tiếp theo sau khi gọi nó và có cơ chế thông báo về Events của Node.js giúp Server nhận được phản hồi từ các API gọi trước đó.

- **Chạy rất nhanh (Very Fast)** : Dựa trên V8 Javascript Engine của Google Chrome, thư viện Node.js rất nhanh trong các việc thực thi code.

- **Đơn luồng nhưng khả năng mở rộng cao (Single Threaded but Highly Scalable)**: Node.js sử dụng một mô hình đơn luồng với vòng lặp sự kiện. Các cơ chế sự kiện giúp Server trả lại các phản hồi với một cách non-blocking và tạo cho

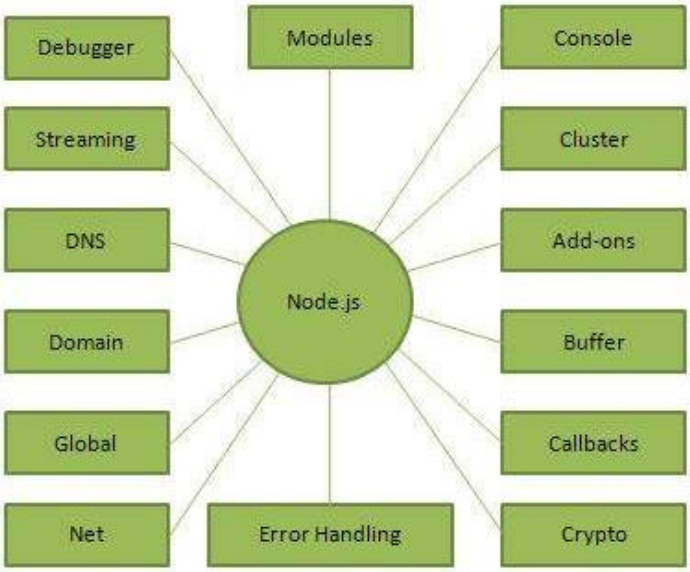
Server khả năng mở rộng khác hẳn với các cách truyền thống tạo ra một số lượng luồng hữu hạn để quản lý request. Nodejs sử dụng các chương trình đơn luồng và các chương trình này cung cấp các dịch vụ cho số lượng request nhiều hơn so với các Server truyền thống như Apache HTTP Server.

- **Không đệm (No Buffering)** : Ứng dụng Node.js không lưu trữ các dữ liệu buffer. Những ứng dụng này đơn giản là in ra dữ liệu dưới dạng các đoạn nhỏ (chunk).

- **Có giấy phép (License)** : Node.js được phát hành dựa vào MIT License.

➢ Các thành phần quan trọng trong Nodejs

Dưới đây là các thành phần quan trọng làm cho Nodejs trở lên mạnh mẽ.



➢ Khi nào nên sử dụng NodeJs ?

- Các ứng dụng I/O bound (phụ thuộc vào hệ thống I/O)
- Các ứng dựng về luồng dữ liệu
- Các ứng dụng chuyên về dữ liệu hướng đến thời gian thực
- Các ứng dụng dựa vào JSON APIs
- Các ứng dụng Single Page.

➢ Khi nào không nên sử dụng Nodejs?

- Nó không nên sử dụng trong các ứng dụng đòi hỏi về CPU, ví dụ như các công việc tính toán, xử lý dữ liệu lớn, xử lý ảnh, đa luồng …

2. ExpressJS Framework

➢ ExpressJS là gì?

Express là một framework nhỏ và linh hoạt cho các ứng dụng web sử dụng Nodejs, cung cấp một lượng lớn của tính năng mạnh mẽ để phát triển các ứng dụng web và mobile. Nó làm cho việc phát triển các ứng dụng dựa trên Node.js trở nên nhanh chóng, dễ dàng.

➢ Các tính năng cơ bản của Express framework:

- Cho phép thiết lập các Middleware để hồi đáp các HTTP request.
- Định nghĩ bảng định tuyến có thể được sử dụng để thực hiện các hành động khác nhau dựa trên phương thức HTTP và URL.
- Cho phép trả về các trang HTML dựa vào các tham số truyền vào đến template một cách linh động.

3. MongoDB

➢ MongoDB là gì?

MongoDB là một cơ sở dữ liệu mã nguồn mở và là cơ sở dữ liệu NoSQL hàng đầu, được viết bằng C++.

MongoDB là một cơ sở dữ liệu đa nền tảng, hướng tài liệu (document), cung cấp hiệu năng cao, khả năng sẵn dụng cao và khả năng dễ dàng mở rộng.

➢ Một số khái nhiệm cơ bản trong MongoDB:

- Khái niệm *Database*: *Database* là một nơi chứa vật lý cho các *Collection*. Mỗi *Database* chứa một tập hợp các file riêng của nó trên hệ thống file. Mỗi *MongoDB Server* chứa nhiều *Database*.
- Khái niệm *Collection*: *Collection* là một nhóm các MongoDB *Document*. Nó tương đương như một bảng trong RDBMS (Hệ cơ sở dữ liệu quan hệ). Do đó, một *Collection* chỉ tồn tại bên trong một *Database*. Các *Document* bên trong một *Collection* có thể có các trường khác nhau. Nói chung, tất cả các *Document* trong một *Collection* là tương tự nhau hoặc với cùng mục đích.
- Khái niệm *Document*: Một *Document* trong MongoDB là một tập hợp các cặp *key-value* (hoặc *field - value*). Các *Document* có *schema* động, nghĩa là *Document* trong cùng một *Collection* không cần thiết phải có cùng một tập hợp các trường hoặc cấu trúc giống nhau, và các trường giống nhau giữa các *Document* của một *Collection* có thể chứa giá trị có kiểu dữ liệu khác nhau.

➢ MongooseJS là gì?

Mongoose là một công cụ mô hình hóa đối tượng MongoDB, được thiết kế để hoạt động trong môi trường không đồng bộ.

Mongoose cung cấp một giải pháp đơn giản dựa trên Schema để mô hình hóa dữ liệu trong ứng dụng Nodejs. Nó bao gồm các tiện ích như ép kiểu định sẵn, xác nhận hợp lệ, xây dựng truy vấn, các móc nối logic nghiệp vụ và nhiều tác vụ khác.

4. Python-shell

➢ Python-shell là gì?

- Là một thư viện cung cấp cho ta một cách đơn giản để chạy các tập lệnh Python từ Node.js với sự giao tiếp giữa các tiến trình cơ bản nhưng hiệu quả và xử lý lỗi tốt hơn.

➢ Các đặc trưng của Python-shell

- Chạy các mã lệnh Python trong một child process một cách đáng tin cậy. (vì python-shell xây dựng dựa trên module child_process của NodeJS do đó nó tận dụng được các khả năng của module này, có thể tìm hiểu thêm tại đây)
- Được tích hợp sẵn các chế độ giao tiếp dữ liệu như: text, JSON hay binary.
- Có bộ phân tích cú pháp (parser) và định dạng (formatter) tuỳ chỉnh.
- Truyền dữ liệu đơn giản và hiệu quả qua các dòng stdin và stdout
- Stack trace được mở rộng khi một lỗi được ném ra

II. Front-end

- *Về phía client, ngoài các công nghệ bắt buộc như HTML và CSS, nhóm lựa chọn sử dụng AngularJS trong đó đi kèm với Bootstrap, JQuery.*

1. Bootstrap and jQuery

➢ Bootstrap là gì?

- *Bootstrap là một framework CSS được Twitter phát triển. Nó là một tập hợp các bộ chọn, thuộc tính và giá trị có sẵn để giúp web designer tránh việc lặp đi lặp lại trong quá trình tạo ra các class CSS và những đoạn mã HTML giống nhau trong dự án web của mình.*

➢ JQuery là gì?

- *jQuery là một thư viện JavaScript nhanh và ngắn gọn được tạo bởi John Resig vào năm 2006 với một phương châm:* ***Write less, do more.***
- *jQuery là một bộ công cụ JavaScript để đơn giản hóa rất nhiều tác vụ với lượng code phải viết ít hơn. Sau đây là danh các đặc trưng chủ yếu được hỗ trợ bởi jQuery:*

  ✓ **Thao tác với DOM (DOM manipulation)** – jQuery làm cho việc lựa chọn các phần tử DOM, di chuyển giữa chúng và chỉnh sửa nội dung của

chúng trở lên dễ dàng bằng cách sử dụng một bộ máy lựa chọn mã nguồn mở cross-browser gọi là **Sizzle**.

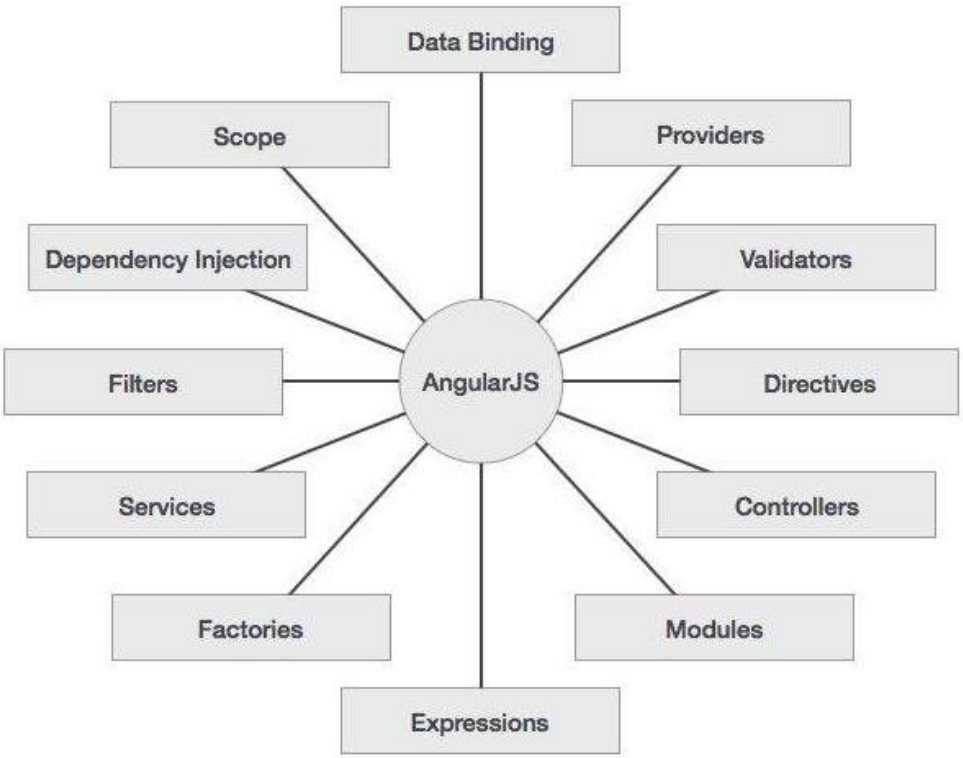
- ✓ **Kiểm soát sự kiện (Event handling)** – jQuery đề xuất một cách tao nhã hơn để bắt được một tập rất lớn các sự kiện.
- ✓ **Hỗ trợ AJAX (AJAX Support)** – jQuery giúp bạn rất nhiều trong việc phát triển một website responsive và chức năng phong phú bằng cách sử dụng công nghệ Ajax.
- ✓ **Chuyển động (Animations)** – jQuery có rất nhiều các hiệu ứng chuyển động được tích hợp sẵn mà bạn có thể sử dụng làm website trở nên sinh động hơn.
- ✓ **Nhẹ (Lightweight)** – chỉ nặng 19KB
- ✓ **Hỗ trợ đa trình duyệt (Cross Browser Support)** – làm việc tốt trên nhiều trình duyệt khác nhau: IE 6.0+, FF 2.0+, Safari 3.0+, Chrome and Opera 9.0+
- ✓ **Công nghệ mới nhất (Latest Technology)** – jQuery hỗ trợ bộ chọn của CSS3 và cú pháp XPath cơ bản.

2. AngularJS

➢ AngularJS là gì?

- *AngularJS là một framework có cấu trúc cho các ứng dụng web động, cho phép chúng ta sử dụng HTML như là ngôn ngữ mẫu và cho phép mở rộng cú pháp của HTML để diễn đạt các thành phần ứng dụng của bạn một cách rõ ràng và chi tiết nhất. Data binding và depecdency injection của AngularJS loại bỏ một lượng lớn code mà bình thường ta phải viết.*

➢ Những đặc trưng của AngularJS

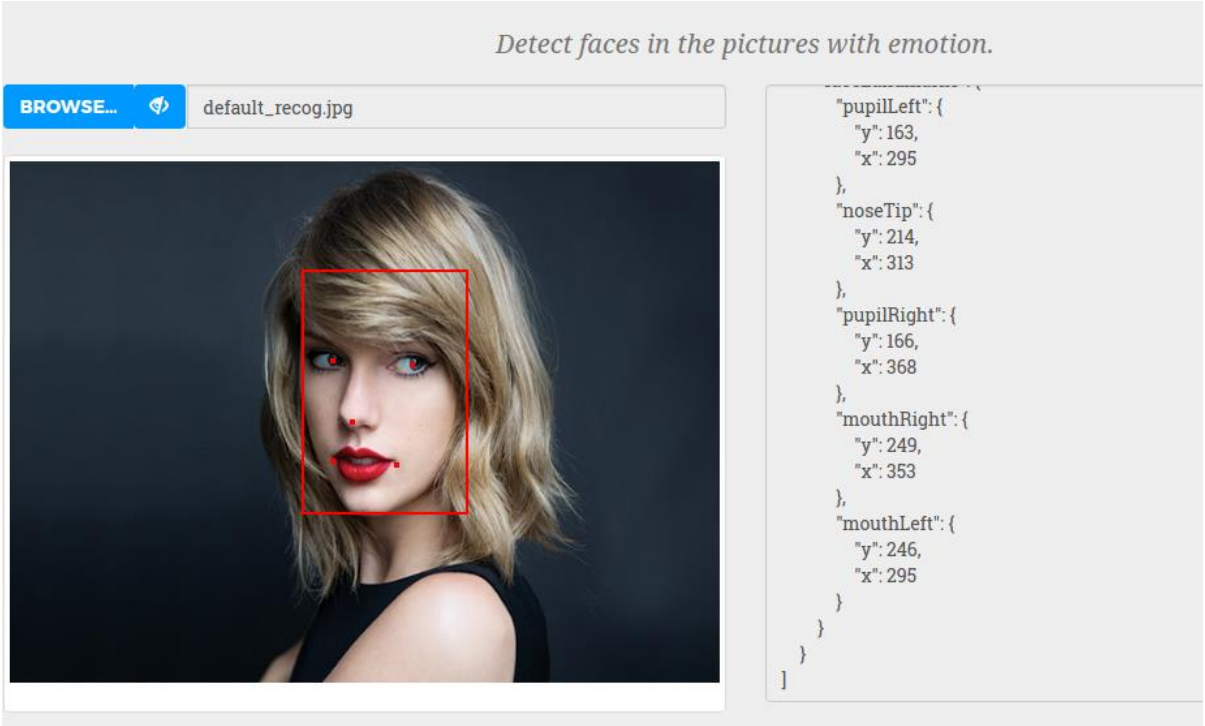


- ***Liên kết dữ liệu (Data-binding):*** *tự động đồng bộ dữ liệu giữa 2 thành phần model và view.*
- ***Biến tầm vực (Scope):*** *Đây là những đối tượng mà trỏ đến model.*
- ***Controller:*** *đây là những hàm JavaScript mà được gắn liền với những scope cụ thể.*
- ***Dịch vụ (Services):*** *AngularJS có dựng sẵn một số dịch vụ ví dụ như $https: để tạo XMLHttpRequests. Đây là các đối tượng đơn nhất chỉ được khởi tạo một lần duy nhất trong một app.*
- ***Bộ lọc (Filters)*** *: chúng chọn một tập con của các phần tử từ một mảng và trả về một mảng mới.*
- ***Chỉ dẫn (Directives):*** *được thêm vào các phần tử của DOM (như elements, attributes, css, ...) như ngBind, ngModel, ... Chúng có thể được sử sụng để tạo ra các thẻ HTML mới tùy chỉnh mà dùng để phục vụ các tiện ích mới.*
- ***Mẫu (Templates):*** *chúng là các bản mẫu cho thành phần view, được in ra với các thông tin từ controller và model.*
- ***Điều hướng (Routing):*** *là khái niệm chuyển đổi giữa các thành phần view.*
- ***Mô hình MVM (Model View Whatever):*** *AngularJS không chỉ cài đặt mô hình MVC theo ý nghĩa truyền thống, mô hình đó gần với Model-View-View-Model (MVVM)*
- ***Kết nối sâu (Deep Linking):*** *cho phép bạn mã hóa trạng thái của ứng dụng trong URL vì vậy nó có thể được đánh dấu bookmark (hay lưu lại) để có thể gọi lại bất cứ lúc nào. Ứng dụng sau đó có thể được trả về trạng thái nào đó từ URL.*
- ***Truyền phụ thuộc (Dependency Injection):*** *AngularJS có một hệ thống con các thành phần phụ thuộc có thể truyền vào các hàm, chúng làm cho ứng dụng trở lên dễ dàng phát triển, dễ hiểu và kiểm tra.*

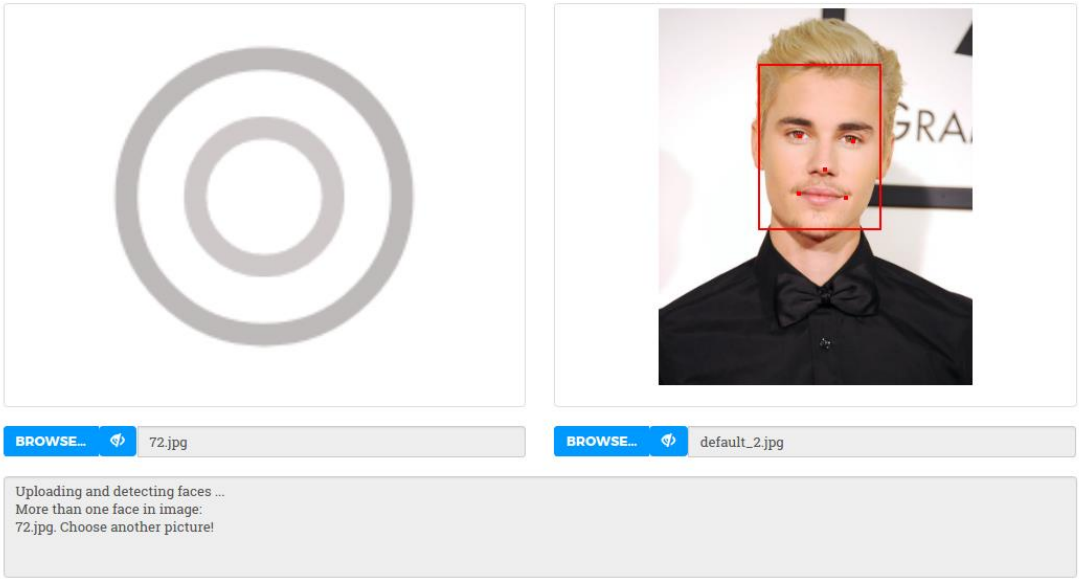
3. Giao diện website

- *Website cung cấp cho người dùng 2 dịch vụ chính là phát hiện khuôn mặt (faces detection) và xác thực các khuôn mặt (faces verification).*
- *Với chức năng faces detection, người dùng có khả năng chọn một ảnh để upload, hệ thống sẽ trả về kết quả nếu ảnh đó có ít nhất một mặt được phát hiện, ngược*

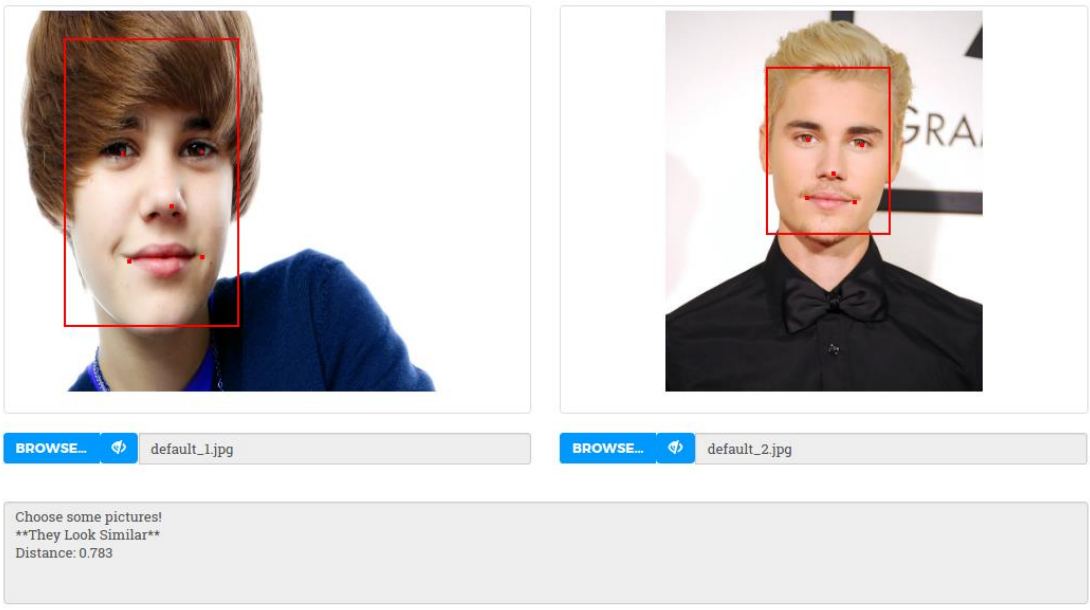
lại nếu hệ thống không tìm thấy mặt nào trong tấm ảnh đó sẽ thông báo người dùng chọn một ảnh khác.

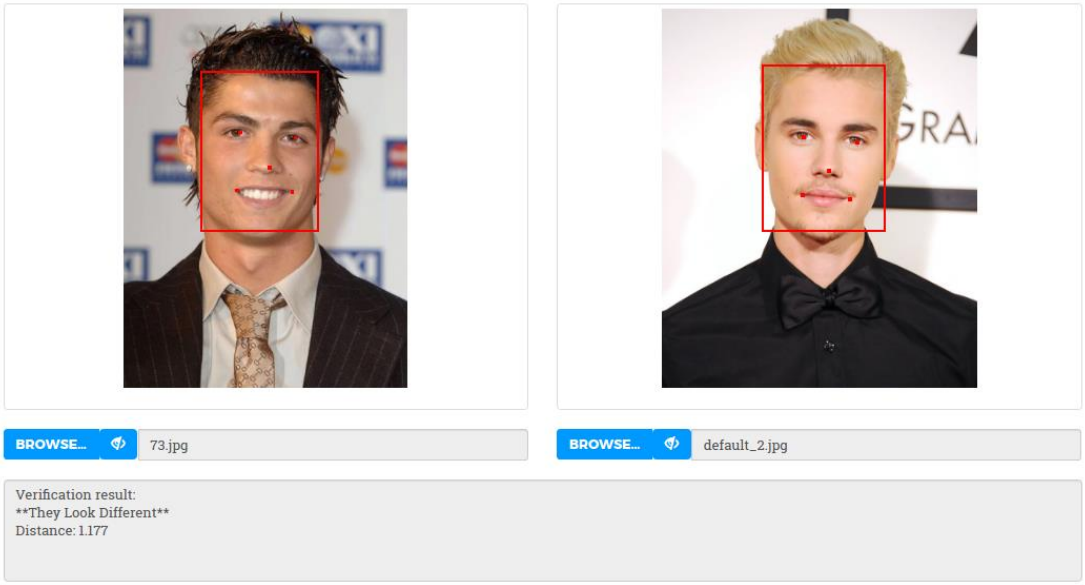


- *Tương tự với chức năng faces verification, người dùng sẽ chọn lần lượt 2 bức ảnh để upload, ngay sau mỗi lần upload, hệ thống sẽ trả về kết quả so sánh hai khuôn mặt. Có 2 trường hợp xảy ra:*
    - *Nếu ảnh người dùng upload không phát hiện được khuôn mặt nào hoặc nhiều hơn 2 khuôn mặt trong một bức ảnh, hệ thống sẽ thông báo cho người dùng chọn ảnh khác và lý do tương ứng.*



- *Nếu ảnh người dùng upload thỏa mãn mỗi bức ảnh chỉ có 1 khuôn mặt được phát hiện, hệ thống sẽ trả về kết quả so sánh họ trông giống nhau hay khác nhau cùng độ khác nhau dựa trên giá trị distance.*


BROWSE...
default_1.jpg
BROWSE...
default_2.jpg
Choose some pictures!
They Look Similar
Distance: 0.783


BROWSE...
73.jpg
BROWSE...
default_2.jpg
Verification result:
They Look Different
Distance: 1.177

TÀI LIỆU THAM KHẢO

- FaceNet: A Unified Embedding for Face Recognition and Clustering
- A Discriminative Feature Learning Approach for Deep Face Recognition
- Multi-task CNN
- https://www.tutorialspoint.com
- http://vietjack.com
- https://docs.mongodb.com