

# MÔ HÌNH FACENET

Người làm: Châu Trường Long

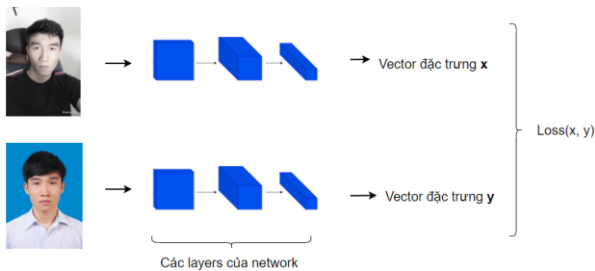
**Giới thiệu về facenet:** Facenet là một mô hình học máy deep learning trong việc ứng dụng các bài toán nhận diện nói chung và nhận diện gương mặt nói riêng. Thường ứng dụng cho các bài toán như chấm công, điểm danh, hay trong bảo mật, ... Cốt lõi của facenet được tạo nên từ ba thành phần chính rất quan trọng đó là *Siam network* và *triple loss function* và *PCA (Principal Component Analysis)*.

## 1. Siam network

Siam network với tên đầy đủ là Siamese Neural Network, dựa trên một mạng nơ-ron tích chập (CNN) đã lược bỏ đi output layer. Kiến trúc mạng này sinh ra với mục đích giải quyết bài toán liệu từ hai bức ảnh chứa mặt người chụp trong hoàn cảnh khác nhau thì có phải cùng một người không.

Siam network được ứng dụng cho các bài toán thực tế như: điểm danh, chấm công, hay mở chức năng mở khóa điện thoại bằng gương mặt, ... Siam network được giới thiệu lần đầu trong bài báo [1] "DeepFace: Closing the Gap to Human-Level - Yaniv Taigman, et."

**Input** của mạng Siam là hai bức ảnh bất kỳ, **Output** là hai vector đặc trưng của bức ảnh. Mà vector đặc trưng biểu diễn ở dạng số học nên có thể sử dụng chúng để kiểm tra xem độ tương đồng giữa các bức ảnh bằng cách đưa vào một loss function. Hàm loss thường được dùng là một norm bậc 2 [2].



Công thức Norm bậc 2 có dạng:  $\|f(x_1) - f(x_2)\|_2^2$

Vì vậy nếu hai ảnh đầu vào là cùng một người thì loss function sẽ rất **nhỏ**. Ngược lại nếu hai ảnh đầu vào không cùng một người thì loss function sẽ **lớn**.

Không nhất thiết phải sử dụng loss function là norm 2. Có thể chọn một loss function khác và tất nhiên nó cũng ảnh hưởng đến kết quả.

## 2. Facenet

Sở dĩ phải nhắc đến Siam network trong model facenet, bởi vì facenet chính là một dạng Siam network.

Giống với Siam network, facenet sẽ tìm cách biểu diễn một bức ảnh dưới dạng một vector đặc trưng thông qua một bottleneck layer (layer nghẽn cổ chai).

Vậy bottleneck layer là gì? Giống với tên gọi của nó, mọi thứ đi qua nó đều phải nhỏ lại. Mục đích là để

giảm chiều dữ liệu của các vector đặc trưng này lại. Tại sao phải giảm chiều của dữ liệu? Vì chiều của dữ liệu tương đối lớn, nên việc tính toán rất chậm ảnh hưởng đến cả một hệ thống lớn. Cho nên giải pháp là phải sử dụng một thuật toán giảm chiều dữ liệu gọi là **PCA**.

Một hạn chế khác là việc lựa chọn loss function cũng ảnh hưởng đến kết quả rất nhiều. Các loss function bình thường như norm bậc hai **chỉ học được một trong hai** khả năng xảy ra đó là khác nhau hoặc giống nhau, bởi vì các loss function này chỉ đo lường khoảng cách giữa hai điểm. Giải pháp của vấn đề này là sử dụng **triple loss function**.

## 3. Triple loss

Triple loss sử dụng một khái niệm gọi là **anchor**. Cụ thể trong hệ thống này sẽ là **anchor image**. Anchor image này sẽ là các bức ảnh gốc của sinh viên như ảnh thẻ 3x4, nói chung bức ảnh này là cố định.

Như tên của loss function này là triple nên sẽ cần đến 3 bức ảnh: một là anchor, hai là positive và ba là negative.



Positive

Anchor

Negative

Mục tiêu là tối thiểu hóa khoảng cách giữa 2 ảnh khi chúng là negative và tối đa hóa khoảng cách khi chúng là positive. Thật ra có thể hiểu đơn giản là triple loss có thể kết hợp từ hai norm loss function. Do đó công thức là:

$$\mathcal{L}(A, P, N) = \sum_{i=0}^n \max(\|f(A_i) - f(P_i)\|_2^2 - \|f(A_i) - f(N_i)\|_2^2 + \alpha, 0)$$

Do đó khi đưa lần lượt bộ ba hình ảnh vào train model thì model sẽ học được cả trường hợp sai lẫn đúng.

## 4. Principal Component Analysis (PCA)

PCA là một thuật toán quan trọng trong mô hình facenet. Nó giúp cho việc tính toán trở nên nhanh

chóng nhờ vào công việc duy nhất đó là giảm chiều dữ liệu. Bằng cách loại bỏ các thành phần không quan trọng trong một bức ảnh, chỉ giữ lại các nét đặc trưng quan trọng của bức ảnh.

Các vector đặc trưng trong facenet thường có số chiều rất lớn (>1000), nhưng khi đi qua thuật toán PCA chúng chỉ còn khoảng 128 chiều. Thuật toán PCA rất phức tạp nó vận dụng nhiều kiến thức về toán học và đại số tuyến tính, đặc biệt là ma trận. Bài viết này chỉ nói qua về cách làm việc và tác dụng của PCA, chi tiết thuật toán đây [5].

## 5. Pretrain model của David Sandberg

### Pre-trained models

Model name	LFW accuracy	Training dataset	Architecture
<a href="#">20180408-102900</a>	0.9905	CASIA-WebFace	<a href="#">Inception ResNet v1</a>
<a href="#">20180402-114759</a>	0.9965	VGGFace2	<a href="#">Inception ResNet v1</a>

NOTE: If you use any of the models, please do not forget to give proper credit to those providing the training dataset as well.

- Link: <https://github.com/davidsandberg/facenet>
- Hai pretrain model facenet có chính xác rất cao. Vì vậy trước tiên sẽ triển khai hệ thống theo model này. Sau đó sẽ đi xây dựng một model riêng.

### Tài liệu tham khảo

- [1] Yaniv Taigman, “DeepFace: Closing the Gap to Human-Level - [Yaniv Taigman. elt.](#)”
- [2] Norm, “<https://machinelearningcoban.com/math/>”
- [3] Phạm Đình Khánh, “[Mô hình Facenet trong face recognition](#)”
- [4] Phạm Đình Khánh, “[Thực hành training Facenet](#)”
- [5] PCA, “[Principal Component Analysis](#)”