Bài 27 - Mô hình Facenet trong face recognition

12 Mar 2020 - phamdinhkhanh

Menu

1. Hệ thống nhận diện khuôn mặt

1.1. Các ứng dụng phổ biến

Vào năm 2012, tôi vẫn còn nhớ facebook bắt dầu có chức năng nhận diện khuôn mặt. Khi đó tôi còn là một sinh viên chưa biết đến khái niệm trí tuệ nhân tạo. Tôi cực kì ấn tượng với khả năng kì diệu này. Tôi nghĩ chắc facebook phải thuê hàng ngàn người ngồi gán nhãn cho từng bức ảnh mỗi khi người dùng update lên. Hơi ngây thơ chút, nhưng đó là tất cả những gì tôi có thể biết cách đây 8 năm.

Vào năm 2018, apple bắt đầu tích hợp chức năng nhận diện khuôn mặt trong các dòng sản phẩm iphone X. Sau đó, một xu hướng các smart phone ứng dụng nhận diện khuôn mặt ra đời.

Một số ngân hàng ở Autralia bắt đầu ứng dụng xác thực khuôn mặt trong các giao dịch ATM.

Tại Trung Quốc, hệ thống nhận diện khuôn mặt được triển khai trên toàn quốc giúp chấm điểm công dân và đồng thời xác minh nhiều tội phạm lẩn trốn.

Còn tại cơ quan tôi, hệ thống nhận diện khuôn mặt được nghiên cứu in-house áp dụng để chấm công cho nhân viên.

Tôi nghĩ rằng nhận diện khuôn mặt có tính ứng dụng cao. Rất nhiều công ty, doanh nghiệp và quốc gia đang cần.

Tuy nhiên khá kì lạ là nhiều nơi chưa áp dụng nhận diện khuôn mặt vào cơ quan của mình. Việc xây dựng một hệ thống nhận diện khuôn mặt là không hề khó. Hãy có niềm tin rằng bạn có thể tự xây dựng một hệ thống nhận diện khuôn mặt cho cơ quan của mình sau bài viết này.

Top

Trước tiên tôi sẽ viết về lý thuyết của thuật toán. Phần thực hành tôi sẽ giới thiệu ở một bài khác.

Đối với các bạn muốn hiểu sâu về thuật toán có thể bắt đầu đọc bài này.

Đối với các bạn muốn thấy trước hiệu quả. Hãy chờ bài thực hành sau bài này.

1.2. Các loại hệ thống xác thực

Hầu hết các hệ thống xác thực sẽ dựa trên thông tin sinh trắc để định danh một người. Các thông tin sinh trắc là những thông tin duy nhất với mỗi người. Đó có thể là vân tay, ADN, vân mắt, khuôn mặt (không phẫu thuật thẩm mỹ, hãy thả lỏng khuôn mặt nhất có thể, không chu môi, nháy mắt, tạo kiểu cute các thứ), hình dạng, dáng đi,….

Theo đó, có vô số các phương pháp xác thực khác nhau, tùy vào việc sử dụng thông tin sinh trắc nào. Ở Việt Nam thì chắc phổ biến nhất là xác thực bằng vân tay.

Trước đây khi computer vision chưa phát triển thì các hệ thống xác thực bằng vân tay, bằng vân mắt được áp dụng khá phổ biến. Vài năm gần đây, xác thực bằng khuôn mặt mới được áp dụng rộng rãi. Xét về độ chính xác thì các hệ thống xác thực vân tay, vân mắt đáng tin cậy hơn so với nhận diện khuôn mặt vì dữ liệu không thay đổi, trong khi khuôn mặt một người có thể thay đổi qua thời gian và chịu sự co dãn.

Tuy nhiên hệ thống nhận diện khuôn mặt được ưa chuộng vì quá trình xác thực nhanh, có thể xác thực từ xa, không cần phải tiếp xúc như vân tay, vân mắt. Bạn được phép nhìn người khác nhưng muốn động chạm thì cần phải được sự xin phép của họ.

1.3. Các phương pháp xác thực khuôn mặt

1.3.1. Phương pháp truyền thống

Các phương pháp truyền thống: Xác thực khuôn mặt bằng cách trích xuất ra một land mark cho face. Land mark như là một bản đồ xác định các vị trí cố định trên khuôn mặt của một người như mắt, mũi, miệng, lông mày,….

Như vậy thay land mark face đã loại bỏ những phần thông tin không cần thiết và giữ lại những thông tin chính. Khi đó mỗi khuôn mặt sẽ được nén thành một véc tơ n chiều. Thông thường là 68 chiều.

Sử dụng đầu vào là land mark face, áp dụng các thuật toán cổ điển như SVM, k-NN, Naive Bayes, Random Forest, MLP, … để phân loại khuôn mặt cho một người.

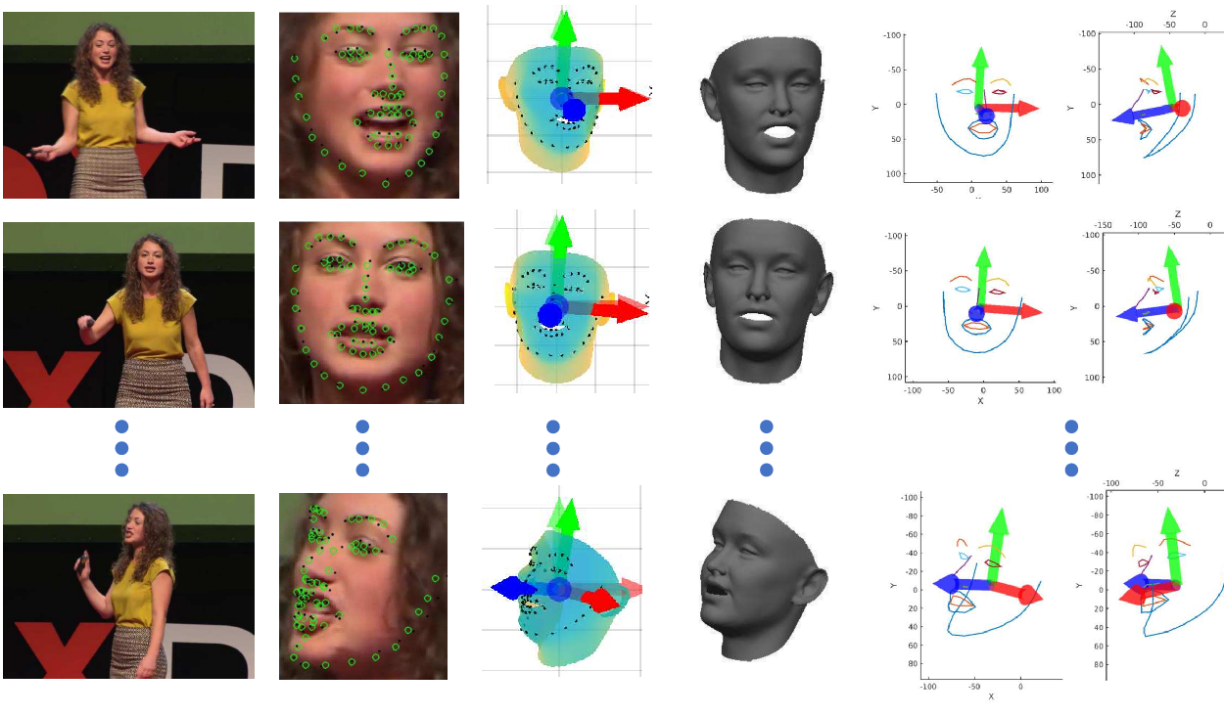
1.3.2. nhận diện 3D

Kĩ thuật nhận diện 3D sẽ sử dụng không gian 3 chiều để biểu diễn khuôn mặt. Từ thông tin này có thể xác định các đặc trưng khác nhau trên bề mặt khuôn mặt như các đường countour của mắt, mũi, cằm.

Một lợi thế của nhận diện khuôn mặt 3D là không bị ảnh hưởng bởi những thay đổi về ánh sáng như các phương pháp 2D. Dữ liệu 3D đã cải thiện đáng kể độ chính xác của nhận dạng khuôn mặt.

Top

Để tạo ra một ảnh 3D, một cụm ba camera được áp dụng. Mỗi camera sẽ hướng vào một góc khác nhau. Tất cả các camera này phối hợp cùng nhau trong việc theo dõi khuôn mặt của một người trong thời gian thực và có thể nhận diện chúng.



(a) Input video (b) Landmark localisation (c) Camera estimation (rigid SfM) (d) Dense 3D shape estimation (e) Sampling of 3D shape on face landmarks

Nhận diện khuôn mặt của iphone là nhận diện 3D. Bạn sẽ phải quay tròn khuôn mặt của mình khi xác thực nó để thuật toán học các góc độ khác nhau.

1.3.3. Các phương pháp nhận diện khác

Ngoài ra còn có một số phương pháp nhận diện khuôn mặt như nhận diện cảm biến da và phương pháp kết hợp.

Phương pháp kết hợp có thể sử dụng nhiều thông tin từ đồng thời land mark face, nhận diện 3D, nhận diện cảm biến da và mang lại độ chính xác cao hơn vì nó nhận diện tốt được trong các trường hợp khuôn mặt có các biểu cảm như cau mày, chớp mắt, co dãn khi cười, nói và ít nhạy cảm với chiếu sáng.

2. Các bài toán khác nhau về face

Có nhiều lớp bài toán khác nhau liên quan đến dữ liệu face. Một trong số 4 bài toán phổ biến nhất dựa trên nhu cầu thực tế cần áp dụng đó là:

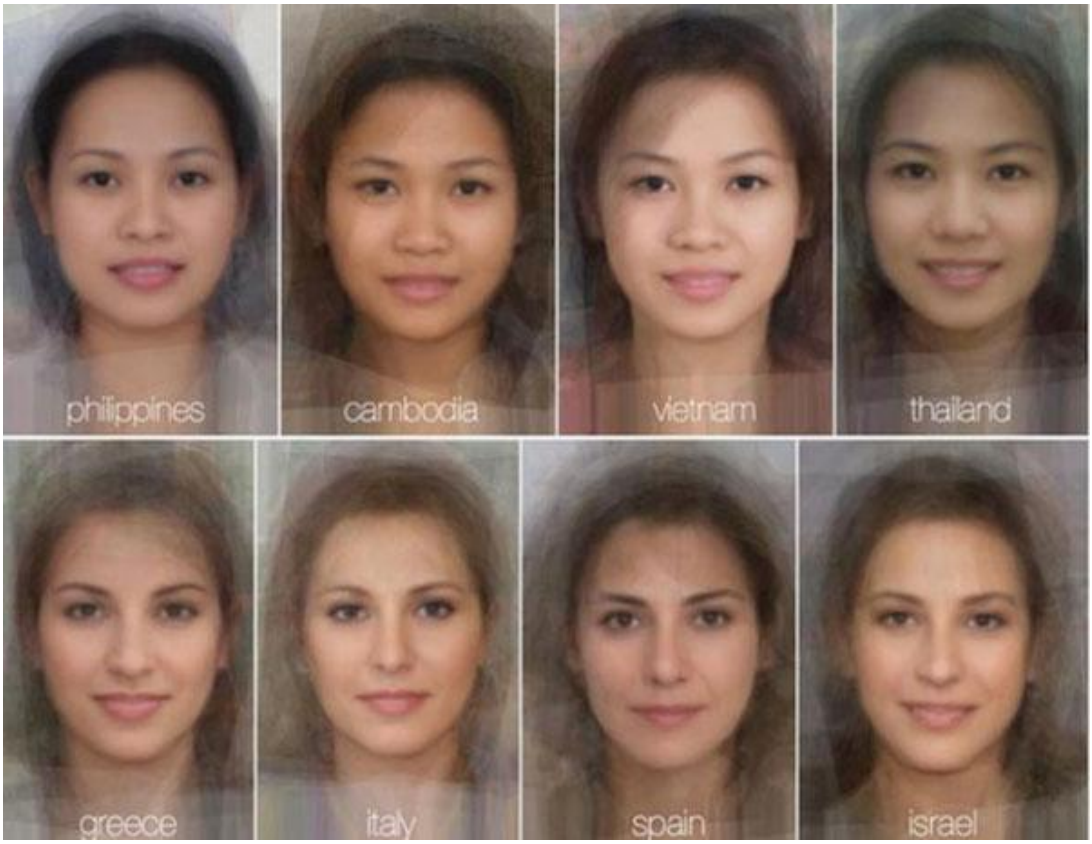
- Nhận diện khuôn mặt (face identification): Đây là bài toán match one-many. Bài toán này sẽ trả lời cho câu hỏi “người này là ai?” bằng cách nhận input là ảnh khuôn mặt và output là nhãn tên người trong ảnh. Tác vụ này thường được áp dụng trong các hệ thống chấm công, hệ thống giám sát công dân, hệ thống cammera thông minh tại các đô thị.
- Xác thực khuôn mặt (face verification): Đây là bài toán match one-one. Bài toán này trả lời cho câu hỏi “có phải 2 ảnh đầu vào là cùng một người không?” Kết quả output sẽ là yes hoặc no. Bài toán thường được dùng trong các hệ thống bảo mật. Xác thực khuôn mặt trên điện thoại của bạn là một bài toán như vậy.

Do có mối quan hệ khá gần nên face recognition là tên gọi chung cho cả hai thuật toán face identification và face verification.

- Tìm kiếm khuôn mặt đại diện (face clustering): Chắc hẳn bạn đã từng đọc hoặc xem video về người có khuôn mặt đặc trưng nhất thế giới (https://www.youtube.com/watch?v=xLaDbe7P2RU). Đơn giản là ta chỉ cần tính ra trung bình của các ảnh khuôn mặt để thu được centroid image. Tính similarity giữa centroid với toàn bộ khuôn mặt còn lại để thu được khuôn mặt đặc trưng nhất similarity nhất với

Top

centroid. Tương tự như vậy bạn cũng có thể tìm ra khuôn mặt đặc trưng nhất của nam, nữ các quốc gia.



- Tìm kiếm khuôn mặt tương đương (face similarity): Chắc các bạn đã từng nghịch qua một số ứng dụng của facebook như bạn giống diễn viên điện ảnh nào nhất. Thuật toán này khá đơn giản, chỉ cần đo lường ảnh mà bạn upload lên với các ảnh diễn viên sẵn có và chọn ra một cái gần nhất. Nhưng đừng tin là mình là họ nhé. Có thể đây chỉ là 1 hình ảnh hai số phận.

3. Các thuật toán nhận diện khuôn mặt

3.1. One-shot learning

One-shot learning là thuật toán học có giám sát mà mỗi một người chỉ cần 1 vài, rất ít hoặc thậm chí chỉ 1 bức ảnh duy nhất (để khởi tạo ra nhiều biến thể).

Từ đầu vào là bức ảnh của một người, chúng ta sử dụng một kiến trúc thuật toán CNN đơn giản để dự báo người đó là ai.

Tuy nhiên nhược điểm của phương pháp này là chúng ta phải huấn luyện lại thuật toán thường xuyên khi xuất hiện thêm một người mới vì shape của output thay đổi tăng lên 1. Rõ ràng là không tốt đối với các hệ thống nhận diện khuôn mặt của một công ty vì số lượng người luôn biến động theo thời gian.

Để khắc phục được vấn đề này, chúng ta sử dụng phương pháp learning similarity.

3.2. Learning similarity

Phương pháp này dựa trên một phép đo khoảng cách giữa 2 bức ảnh, thông thường là các norm chuẩn l_1 hoặc l_2 sao cho nếu 2 bức ảnh thuộc cùng một người thì khoảng cách là nhỏ nhất và nếu không thuộc thì khoảng cách sẽ lớn hơn.

$$\begin{cases} d(\text{img1}, \text{img2}) \leq \tau & \rightarrow \quad \text{same} \\ d(\text{img1}, \text{img2}) > \tau & \rightarrow \quad \text{different} \end{cases}$$

Top

Để cụ thể hơn, tôi minh họa qua hình bên dưới:

Hình 1: Phương pháp learning similarity. Thay vì dự báo một phân phối xác suất để tìm ra nhãn phù hợp nhất với ảnh đầu vào. Thuật toán sẽ so sánh khoảng cách giữa ảnh đầu vào (bên phải) với toàn bộ các ảnh còn lại (bên trái). Ta cần chọn một ngưỡng threshold để quyết định ảnh là giống hoặc khác. Giả sử ngưỡng threshold là 0.5. Trong các bức ảnh bên trái thì bức ảnh ở giữa có khoảng cách với ảnh bên phải nhỏ hơn 0.5. Do đó nó được dự báo cùng một người với ảnh bên phải.

Learning similarity có thể trả ra nhiều hơn một ảnh là cùng loại với ảnh đầu vào tùy theo ngưỡng threshold.

Ngoài ra phương pháp này không bị phụ thuộc vào số lượng classes. Do đó không cần phải huấn luyện lại khi xuất hiện class mới.

Điểm mấu chốt là cần xây dựng được một model encoding đủ tốt để chiếu các bức ảnh lên một không gian eucledean n chiều. Sau đó sử dụng khoảng cách để quyết định nhãn của chúng.

Như vậy learning similarity có ưu điểm hơn so với one-shot learning khi không phải huấn luyện lại model khi mà vẫn tìm ra được ảnh tương đồng.

Vậy làm thế nào để học được biểu diễn của ảnh trong không gian euledean n chiều? Kiến trúc siam network sẽ giúp chúng ta thực hiện điều này một cách dễ dàng.

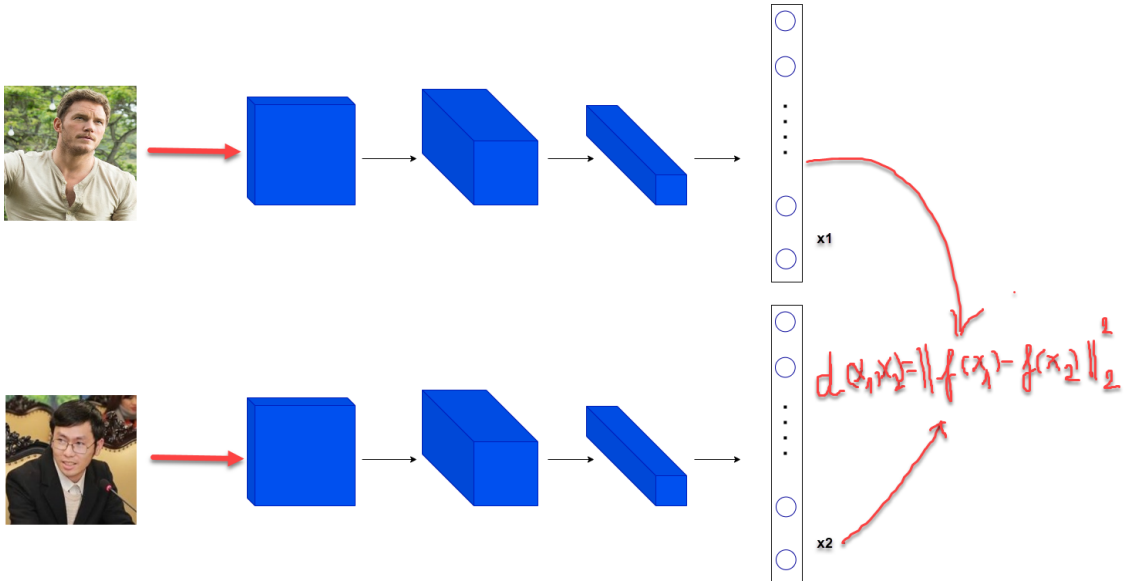
3.3. Siam network

Top

Những kiến trúc mạng mà khi bạn đưa vào 2 bức ảnh và mô hình sẽ trả lời chúng thuộc về cùng 1 người hay không được gọi chung là Siam network. Siam network được giới thiệu đầu tiên bởi DeepFace: Closing the Gap to Human-Level - Yaniv Taigman. elt (https://www.cs.toronto.edu/~ranzato/publications/taigman_cvpr14.pdf).

Kiến trúc của Siam network dựa trên base network là một Convolutional neural network **đã được loại bỏ output layer** có tác dụng encoding ảnh thành véc tơ embedding. Đầu vào của mạng siam network là 2 bức ảnh bất kì được lựa chọn ngẫu nhiên từ dữ liệu ảnh. Output của Siam network là 2 véc tơ tương ứng với biểu diễn của 2 ảnh input. Sau đó chúng ta đưa 2 véc tơ vào hàm loss function để đo lường sự khác biệt giữa chúng. Thông thường hàm loss function là một hàm norm chuẩn bậc 2.

Lý thuyết về norm chuẩn bậc 2 tôi khuyến nghị các bạn xem thêm tại norms chuẩn - machinelearningcoban (https://machinelearningcoban.com/math/#-norms-chuan). Rất đầy đủ và chi tiết.



Hình 2: Từ mô hình Convolutional neural network, mô hình trả ra 2 véc tơ encoding là $\mathbf{x_1}$ và $\mathbf{x_2}$ biểu diễn cho lần lượt ảnh 1 và 2. $\mathbf{x_1}$ và $\mathbf{x_2}$ có cùng số chiều. Hàm $f(\mathbf{x})$ có tác dụng tương tự như một phép biến đổi qua layer fully connected trong mạng neural network để tạo tính phi tuyến và giảm chiều dữ liệu về các kích thước nhỏ. Thông thường là 128 đối đối với các mô hình pretrain.

- Khi $\mathbf{x_1}, \mathbf{x_2}$ là cùng 1 người:

$$||f(\mathbf{x_1}) - f(\mathbf{x_2})||_2^2$$

phải là một giá trị nhỏ.

- Khi $\mathbf{x_1}, \mathbf{x_2}$ là 2 người khác nhau:

$$||f(\mathbf{x_1}) - f(\mathbf{x_2})||_2^2$$

phải là một giá trị lớn.

Khi sử dụng siam network chúng ta sẽ không cần phải lo lắng về vấn đề output shape thay đổi vì base network đã được loại bỏ layer cuối.

Mục tiêu chính của siam network đó là tìm ra biểu diễn của ảnh trong không gian n chiều nên không nhất thiết phải lựa chọn hàm loss function là hàm `binary cross entropy` như các bài toán phân loại nhị phân. Trên thực tế lựa chọn loss function là `binary cross entropy` vẫn tìm ra được biểu diễn tốt cho ảnh. Ngoài ra còn một số các biến thể khác của loss function như sử dụng khoảng cách norm chuẩn l_2 đã chuẩn hóa theo phân phối χ^2 dạng:

$$\frac{||f(\mathbf{x_1}) - f(\mathbf{x_2})||_2^2}{||f(\mathbf{x_1}) + f(\mathbf{x_2})||_2}$$

hoặc sử dụng norm chuẩn l_1.

Top

Việc lựa chọn hàm loss function sẽ có tác động rất lớn tới độ chính xác của biểu diễn ảnh. Ngoài các hàm loss function nêu trên, còn một hàm loss function có hiệu quả rất lớn đối với việc biểu diễn véc tơ cho face. Đó là hàm gì? Chúng ta sẽ tìm hiểu ở thuật toán facenet.

4. Thuật toán facenet

Facenet chính là một dạng siam network có tác dụng biểu diễn các bức ảnh trong một không gian eucledean n chiều (thường là 128) sao cho khoảng cách giữa các véc tơ embedding càng nhỏ, mức độ tương đồng giữa chúng càng lớn.

4.1. Khái quát thuật toán

Hầu hết các thuật toán nhận diện khuôn mặt trước facenet đều tìm cách biểu diễn khuôn mặt bằng một véc tơ embedding thông qua một layer bottle neck có tác dụng giảm chiều dữ liệu.

- Tuy nhiên hạn chế của các thuật toán này đó là số lượng chiều embedding tương đối lớn (thường >= 1000) và ảnh hưởng tới tốc độ của thuật toán. Thường chúng ta phải áp dụng thêm thuật toán PCA để giảm chiều dữ liệu để tăng tốc độ tính toán.
- Hàm loss function chỉ đo lường khoảng cách giữa 2 bức ảnh. Như vậy trong một đầu vào huấn luyện chỉ học được **một trong hai** khả năng là sự giống nhau nếu chúng cùng 1 class hoặc sự khác nhau nếu chúng khác class mà không học được cùng lúc sự giống nhau và khác nhau trên cùng một lượt huấn luyện.

Facenet đã giải quyết cả 2 vấn đề trên bằng các hiệu chỉnh nhỏ nhưng mang lại hiệu quả lớn:

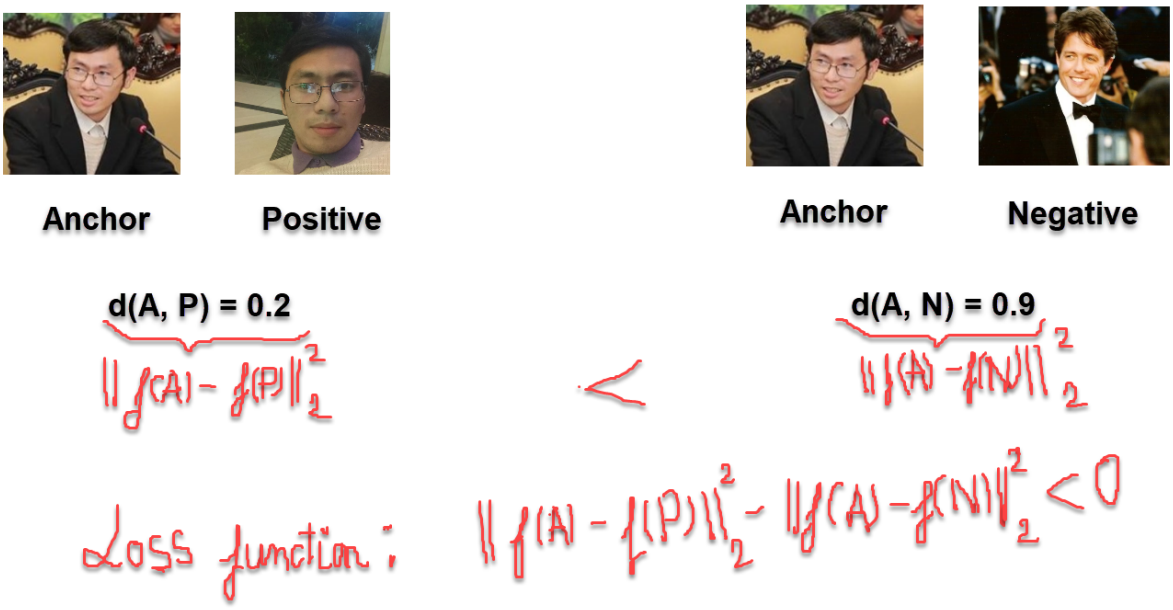
- Base network áp dụng một mạng convolutional neural network và giảm chiều dữ liệu xuống chỉ còn 128 chiều. Do đó quá trình suy diễn và dự báo nhanh hơn và đồng thời độ chính xác vẫn được đảm bảo.
- Sử dụng loss function là hàm triplot loss có khả năng học được **đồng thời** sự giống nhau giữa 2 bức ảnh cùng nhóm và phân biệt các bức ảnh không cùng nhóm. Do đó hiệu quả hơn rất nhiều so với các phương pháp trước đây.

4.2. Triple loss

Trong facenet, quá trình encoding của mạng convolutional neural network đã giúp ta mã hóa bức ảnh về 128 chiều. Sau đó những véc tơ này sẽ làm đầu vào cho hàm loss function đánh giá khoảng cách giữa các véc tơ.

Để áp dụng triple loss, chúng ta cần lấy ra 3 bức ảnh trong đó có một bức ảnh là anchor. Chắc bạn còn nhớ khái niệm về anchor box đã được trình bày tại Bài 25 - YOLO You Only Look Once (https://phamdinhkhanh.github.io/2020/03/09/DarknetAlgorithm.html#5-anchor-box) chứ. Anchor image cũng có tác dụng gần như vậy. Trong 3 ảnh thì ảnh anchor được cố định trước. Chúng ta sẽ lựa chọn 2 ảnh còn lại sao cho một ảnh là negative (của một người khác với anchor) và một ảnh là positive (cùng một người với anchor).

Top



Kí hiệu ảnh Anchor, Positive, Negative lần lượt là $\mathbf{A}, \mathbf{P}, \mathbf{N}$.

Mục tiêu của hàm loss function là **tối thiểu hóa khoảng cách giữa 2 ảnh khi chúng là negative** và **tối đa hóa khoảng cách khi chúng là positive**. Như vậy chúng ta cần lựa chọn các bộ 3 ảnh sao cho:

- Ảnh Anchor và Positive khác nhau nhất: cần lựa chọn để khoảng cách $d(\mathbf{A}, \mathbf{P})$ lớn. Điều này cũng tương tự như bạn lựa chọn một ảnh của mình hồi nhỏ so với hiện tại để thuật toán học khó hơn. Nhưng nếu nhận biết được thì nó sẽ thông minh hơn.
- Ảnh Anchor và Negative giống nhau nhất: cần lựa chọn để khoảng cách $d(\mathbf{A}, \mathbf{N})$ nhỏ. Điều này tương tự như việc thuật toán phân biệt được ảnh của một người anh em giống bạn với bạn.

Triplot loss function luôn lấy 3 bức ảnh làm input và trong mọi trường hợp ta kì vọng:

$$d(\mathbf{A}, \mathbf{P}) < d(\mathbf{A}, \mathbf{N}) \tag{1}$$

Để làm cho khoảng cách giữa vế trái và vế phải lớn hơn, chúng ta sẽ cộng thêm vào vế trái một hệ số α không âm rất nhỏ. Khi đó (1) trở thành:

$$d(\mathbf{A}, \mathbf{P}) + \alpha \leq d(\mathbf{A}, \mathbf{N})$$
$$\rightarrow ||f(\mathbf{A}) - f(\mathbf{P})||_2^2 + \alpha \leq ||f(\mathbf{A}) - f(\mathbf{N})||_2^2$$
$$\rightarrow ||f(\mathbf{A}) - f(\mathbf{P})||_2^2 - ||f(\mathbf{A}) - f(\mathbf{N})||_2^2 + \alpha \leq 0$$

Như vậy hàm loss function sẽ là:

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

Trong đó n là số lượng các bộ 3 hình ảnh được đưa vào huấn luyện.

Sẽ không ảnh hưởng gì nếu ta nhận diện đúng ảnh Negative và Positive là cùng cặp hay khác cặp với Anchor. Mục tiêu của chúng ta là giảm thiểu các trường hợp hợp mô hình nhận diện sai ảnh Negative thành Postive nhất có thể. Do đó để loại bỏ ảnh hưởng của các trường hợp nhận diện đúng Negative và Positive lên hàm loss function. Ta sẽ điều chỉnh giá trị đóng góp của nó vào hàm loss function về 0.

Tức là nếu:

$$||f(\mathbf{A}) - f(\mathbf{P})||_2^2 - ||f(\mathbf{A}) - f(\mathbf{N})||_2^2 + \alpha \leq 0$$

sẽ được điều chỉnh về 0. Khi đó hàm loss function trở thành:

Top

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

Như vậy khi áp dụng Triple loss vào các mô hình convolutional neural network chúng ta có thể tạo ra các biểu diễn véc tơ tốt nhất cho mỗi một bức ảnh. Những biểu diễn véc tơ này sẽ phân biệt tốt các ảnh Negative rất giống ảnh Positive. Và đồng thời các bức ảnh thuộc cùng một label sẽ trở nên gần nhau hơn trong không gian chiếu euclidean.

Một chú ý quan trọng khi huấn luyện mô hình siam network với triplot function đó là chúng ta luôn phải xác định trước cặp ảnh $(\mathbf{A}, \mathbf{P})$ thuộc về cùng một người. Ảnh $\mathbf{N}$ sẽ được lựa chọn ngẫu nhiên từ các bức ảnh thuộc các nhãn còn lại. Do đó cần thu thập ít nhất 2 bức ảnh/1 người để có thể chuẩn bị được dữ liệu huấn luyện.

4.2. Lựa chọn triple images input

Nếu lựa chọn triple input một cách ngẫu nhiên có thể ảnh khiến cho bất đẳng thức (1) dễ dàng xảy ra vì trong các ảnh ngẫu nhiên, khả năng giống nhau giữa 2 ảnh là rất khó. Hầu hết các trường hợp đều thỏa mãn bất đẳng thức (1) và không gây ảnh hưởng đến giá trị của loss function do giá trị của chúng được set về 0. Như vậy việc học những bức ảnh Negative quá khác biệt với Anchor sẽ không có nhiều ý nghĩa.

Để mô hình khó học hơn và đồng thời cũng giúp mô hình phân biệt chuẩn xác hơn mức độ giống và khác nhau giữa các khuôn mặt, chúng ta cần lựa chọn các input theo bộ 3 khó học (hard triplets).

Ý tưởng là chúng ta cần tìm ra bộ ba $(\mathbf{A}, \mathbf{N}, \mathbf{P})$ sao cho (1) là gần đạt được đẳng thức (xảy ra dấu =) nhất. Tức là $d(\mathbf{A}, \mathbf{P})$ lớn nhất và $d(\mathbf{A}, \mathbf{N})$ nhỏ nhất. Hay nói cách khác với mỗi Anchor $\mathbf{A}$ cần xác định:

- **Hard Positive:** Bức ảnh Positive có khoảng cách xa nhất với Anchor tương ứng với nghiệm:

$$\text{argmax}_{\mathbf{P}_i}(d(\mathbf{A}_i, \mathbf{P}_i))$$

- **Hard Negative:** Bức ảnh Negative có khoảng cách gần nhất với Anchor tương ứng với nghiệm:

$$\text{argmin}_{\mathbf{N}_j}(d(\mathbf{A}_i, \mathbf{N}_j))$$

Với i, j là nhãn của người trong ảnh.

Việc tính toán các trường hợp `Hard Positive` và `Hard Negative` có thể được thực hiện offline và lưu vào checkpoint hoặc có thể tính toán online trên mỗi mini-batch.

Bạn đọc có thể đọc kĩ hơn tại mục 3.2 - Triplet selection (https://arxiv.org/pdf/1503.03832.pdf) ở bài báo gốc.

Chiến lược lựa chọn Triple images sẽ có ảnh hưởng rất lớn tới chất lượng của mô hình Facenet. Nếu lựa chọn Triplet images tốt, Facenet sẽ hội tụ nhanh hơn và đồng thời kết quả dự báo chuẩn xác hơn. Lựa chọn ngẫu nhiên dễ dẫn tới thuật toán không hội tụ.

5. Các pretrain model Facenet

Hầu hết chúng ta khi xây dựng một thuật toán nhận diện khuôn mặt sẽ không cần phải train lại mô hình facenet mà tận dụng lại các mô hình pretrain sẵn có. Bạn sẽ không cần phải tốn thời gian và công sức nếu không có đủ tài nguyên và dữ liệu. Đó cũng là lý do tôi cho rằng việc xây dựng mô hình nhận diện khuôn mặt ở thời điểm hiện tại rất dễ dàng.

Những mô hình pretrain được huấn luyện trên các dữ liệu lên tới hàng triệu ảnh. Do đó có khả năng mã hóa rất tốt các bức ảnh trên không gian 128 chiều. Việc còn lại của chúng ta là sử dụng lại mô hình, tính toán embedding véc tơ và huấn luyện embedding véc tơ bằng một classifier đơn giản để phân loại classes.

5.1. Một số bộ dữ liệu public về face

- CASIA-WebFace (http://www.cbsr.ia.ac.cn/english/CASIA-WebFace-Database.html): Bộ dữ liệu bao gồm gần 500k ảnh được thu thập từ khoảng 10k người.

Top

- VGGFace2 (https://www.robots.ox.ac.uk/~vgg/data/vgg_face2/): Bộ dữ liệu gồm khoảng 3 triệu ảnh được thu thập từ gần 9k người.

Trên là 2 bộ dữ liệu về face phổ biến nhất, được sử dụng nhiều trong các bài báo và nghiên cứu về face recognition.

- Các bộ dữ liệu từ Trung Quốc cũng khá tốt. Những dữ liệu này được Baidu public để hỗ trợ các nhà nghiên cứu: baidu dataset (https://ai.baidu.com/broad)
- Ngoài ra bạn có thể tìm kiếm các bộ dữ liệu từ google, facebook, flickr và các phòng lab đã được liệt kê tại top 15 free image datasets for facial recognition (https://lionbridge.ai/datasets/5-million-faces-top-15-free-image-datasets-for-facial-recognition/).
- Hiện tại dữ liệu face cho người Việt thì tôi mới chỉ thấy trong cuộc thi nhận diện người nổi tiếng - aivivn (https://www.aivivn.com/contests/2). Bộ dữ liệu này đã được các bạn AI engineer bên Sun* đóng góp cho cuộc thi miễn phí. Tôi rất ghi nhận sự đóng góp của các bạn đối với cộng đồng AI việt. Tôi hi vọng tương lai sẽ có những bộ dữ liệu về face lớn hơn nữa cho người Việt. Tài nguyên ảnh trên mạng khá là nhiều, đặc biệt là dữ liệu về face. Một vài tip web crawler có lẽ sẽ hữu ích.

5.2. Một số pretrain model phổ biến

Các pretrain model cho facenet khá nhiều:

- Bạn có thể sử dụng pretrain model từ facenet repo - davidsandberg (https://github.com/davidsandberg/facenet)

Pre-trained models

Model name	LFW accuracy	Training dataset	Architecture
20180408-102900	0.9905	CASIA-WebFace	Inception ResNet v1
20180402-114759	0.9965	VGGFace2	Inception ResNet v1

Kiến trúc mà tác giả sử dụng là Inception ResNetv1 trên 2 bộ dữ liệu là CASIA-WebFace và VGGFace2

- Ngoài ra còn rất nhiều các pretrain model khác nằm rải rác trên các nguồn khác nhau. Bạn có thể xem thêm từ chia sẻ Khai báo pretrained model nhận diện người nổi tiếng - aivivn (https://forum.machinelearningcoban.com/t/khai-bao-pretrained-models-cho-nhan-dien-nguoi-noi-tieng/4566).

6. Tổng kết

Như vậy qua bài viết này bạn đã hiểu được phương pháp chính của thuật toán Facenet trong nhận diện khuôn mặt. Thuật toán này đã cải thiện được đồng thời độ chính xác và tốc độ nhờ áp dụng output shape thấp và hàm triple loss function hiệu quả.

Mấu chốt để xây dựng một mô hình facenet tốt đó là chiến lược lựa chọn input triple images sao cho chúng là những bộ 3 khó học (hard triplets).

Việc huấn luyện mô hình Facenet có thể tốn rất nhiều tài nguyên của bạn. Thế nên, trừ khi bạn là doanh nghiệp, muốn đầu tư vào mô hình này cho mục đích thương mại thì có thể xây dựng lại mô hình trên dữ liệu mới. Nếu bạn là sinh viên nghèo vượt khó hãy sử dụng lại pretrain model. Các model pretrain hiện tại rất tốt và tôi nghĩ hiệu quả chưa chắc đã thua kém.

Ở bài sau tôi sẽ hướng dẫn bạn xây dựng một ứng dụng chấm công cho cơ quan của bạn.

7. Tài liệu

Top

1. DeepFace: Closing the Gap to Human-Level Performance in Face Verification - Yaniv Taigman .etc (https://www.cs.toronto.edu/~ranzato/publications/taigman_cvpr14.pdf)
2. FaceNet: A Unified Embedding for Face Recognition and Clustering - Florian Schroff .etc (https://arxiv.org/pdf/1503.03832.pdf)
3. FaceNet - Florian Schroff, Dmitry Kalenichenko, James Philbin Google Inc. (http://llcao.net/cu-deeplearning17/pp/class10_FaceNet.pdf)
4. Deep Learning Face Representation by Joint Identification-Verification - Yi Sun, Xiaogang Wang, Xiaoou Tang (https://arxiv.org/abs/1406.4773.pdf)
5. Face Recognition Based on Improved FaceNet Model - Qiuyue Wei etc (https://www.researchgate.net/publication/329893282_Face_Recognition_Based_on_Improved_FaceNet_Model
6. Introduction to FaceNet: A Unified Embedding for Face Recognition and Clustering - Dhairya Kumar (https://medium.com/analytics-vidhya/introduction-to-facenet-a-unified-embedding-for-face-recognition-and-clustering-dbdac8e6f02)

Top