TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN 2**

**TÌM HIỂU VÀ ỨNG DỤNG HỆ THỐNG NHẬN DẠNG MẶT NGƯỜI**

*Người hướng dẫn*: **TS LÊ ANH CƯỜNG**

*Người thực hiện*: **NGÔ TRÚC KHANG – 51403072**

**BÙI LONG THIÊN – 51403395**

Lớp **: 14050301**

Khoá  **: 18**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2019**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN 2**

**TÌM HIỂU VÀ ỨNG DỤNG HỆ THỐNG NHẬN DẠNG MẶT NGƯỜI**

*Người hướng dẫn*: **TS LÊ ANH CƯỜNG**

*Người thực hiện*: **NGÔ TRÚC KHANG – 51403072**

**BÙI LONG THIÊN – 51403395**

Lớp **: 14050301**

Khoá  **: 18**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2019**

LỜI CẢM ƠN

Đây là phần tác giả **tự viết** ngắn gọn, thể hiện sự biết ơn của mình đối với những người đã giúp mình hoàn thành Luận văn/Luận án. Tuyệt đối không sao chép theo mẫu những “lời cảm ơn” đã có.

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của TS Lê Anh Cường. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày tháng năm*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Ngô Trúc Khang*

*Bùi Long Thiên*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

Trình bày tóm tắt vấn đề nghiên cứu, các hướng tiếp cận, cách giải quyết vấn đề và một số kết quả đạt được, những phát hiện cơ bản trong vòng 1 -2 trang.

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc28337390)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc28337391)

[TÓM TẮT iv](#_Toc28337392)

[MỤC LỤC 1](#_Toc28337393)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 3](#_Toc28337394)

[CHƯƠNG 1 – GIỚI THIỆU ONE-SHOT LEARNING 4](#_Toc28337395)

[1.1 Sự phát triển Convolutional Neural Networks và những vấn đề mới 4](#_Toc28337396)

[1.2 One-shot learning 5](#_Toc28337397)

[1.3 Một số khái niệm cần phân biệt 6](#_Toc28337398)

[1.3.1 Điểm khác biệt giữa face verification và face recognition. 6](#_Toc28337399)

[1.3.2 Điểm khác biệt giữa nhận diện khuôn mặt và phát hiện giả mạo khuôn mặt. 6](#_Toc28337400)

[CHƯƠNG 2 – SIAMESE NETWORKS 8](#_Toc28337401)

[2.1 Siamese networks 8](#_Toc28337402)

[2.1.1 Giới thiệu về Siamese network 8](#_Toc28337403)

[2.1.2 Triplet loss 10](#_Toc28337404)

[2.1.2 Triplet Selection 11](#_Toc28337405)

[2.2 FaceNet 12](#_Toc28337406)

[2.2.1 Kiến trúc Inception 13](#_Toc28337407)

[2.2.3 Cách hệ thống nhận diện một đối tượng. 15](#_Toc28337408)

[2.2.2 Quá trình huấn luyện FaceNet 17](#_Toc28337409)

[CHƯƠNG 3 – XÂY DỰNG ỨNG DỤNG NHẬN DẠNG KHUÔN MẶT 21](#_Toc28337410)

**DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT**

**CÁC KÝ HIỆU**

*f Tần số của dòng điện và điện áp (Hz)*

*p Mật độ điện tích khối (C/m3)*

**CÁC CHỮ VIẾT TẮT**

CSTD Công suất tác dụng

MF Máy phát điện

BER Tỷ lệ bít lỗi

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

**DANH MỤC HÌNH**

[Hình 1: Siamese networks đơn giản. 8](#_Toc28337371)

[Hình 2: Ma trận đặc trưng (embedding). 9](#_Toc28337372)

[Hình 3: Quá trình xử lí dữ liệu của một mạng siamese network. 10](#_Toc28337373)

[Hình 4: Triplet loss. 11](#_Toc28337374)

[Hình 5: FaceNet biến đổi hình ảnh đầu vào thành ma trận đặc trưng. 13](#_Toc28337375)

[Hình 6: Inception module. 13](#_Toc28337376)

[Hình 7: Inception module sử dụng các ma trận giảm chiều. 14](#_Toc28337377)

[Hình 8: Bức ảnh cần phân loại. 15](#_Toc28337378)

[Hình 9: Chuyển bức ảnh cần phân loại thành ma trận đặc trưng. 15](#_Toc28337379)

[Hình 10: Tính khoảng cách giữa X và các hình ảnh trong tập dữ liệu. 16](#_Toc28337380)

[Hình 11: Lựa chọn lớp đối tượng mà X thuộc về dựa vào khoảng cách Euclidean. 16](#_Toc28337381)

[Hình 12: Chọn Anchor. 17](#_Toc28337382)

[Hình 13: Chọn Positive Image. 17](#_Toc28337383)

[Hình 14: Chọn Negative image. 18](#_Toc28337384)

[Hình 15: Điều chỉnh các trọng số (weight). 18](#_Toc28337385)

[Hình 16: Lặp lại các bước trên với một anchor khác. 19](#_Toc28337386)

[Hình 17: Lặp lại với một anchor khác. 19](#_Toc28337387)

[Hình 18: Tiếp tục lặp lại bước trên với một anchor khác. 20](#_Toc28337388)

[Hình 19: Kết quả. 20](#_Toc28337389)

**DANH MỤC BẢNG**

[Bảng 3.1 Ví dụ cho chèn bảng 2](#_Toc387689363)

CHƯƠNG 1 – GIỚI THIỆU ONE-SHOT LEARNING

* 1. Sự phát triển Convolutional Neural Networks và những vấn đề mới

Sự xuất hiện của Convolutional Neural Networks (CNN) đã mang đến một cuộc cách mạng lớn trong lĩnh vực thị giác máy tính (Computer vision), mang đến một hiệu năng đáng kinh ngạc, vượt qua những thuật toán khác được sử dụng trong các ứng dụng phân lớp trước đó. Nó mang đến độ chính xác gần như tuyệt đối, nhưng cũng mang lại cho các nhà nghiên cứu một vấn đề đau đầu khác là training data (dữ liệu huấn luyện).

Để có được một mô hình CNN với độ chính xác cao đòi hỏi bạn phải có một bộ dữ liệu huấn luyện đủ lớn (nếu không muốn nói là khổng lồ). Cũng vì lí do đó, một cuộc thi thường niên mang tên ImageNet Large Scale Visual Recognition Challenge (ILSVRC) được khởi xướng từ năm 2010, với mục đích cung cấp cho các nhà nghiên cứu những bộ dữ liệu huấn luyện phục vụ cho công việc nghiên cứu các bài toán phân lớp với CNN. Kết quả là năm 2017, tập dữ liệu thu thập từ cuộc thi năm đó đã xấp xỉ 2 triệu tấm ảnh. Bên cạnh đó, rất nhiều nhà nghiên cứu khác đã rất nổ lực cung cấp cho cộng đồng nghiên cứu những bộ dữ liệu chất lượng khác.

Mặc dù vậy, những bộ dữ liệu huấn luyện mà cộng đồng nghiên cứu nỗ lực xây dựng trong những năm gần đây vẫn còn khá nhỏ bé nếu so với hai kẻ khổng lồ Google và Facebook. Gần đây, Google đã công bố rằng, một hệ thống nhận diện khuôn mặt do họ phát triển đã sử dụng một bộ dữ liệu gồm 260 triệu tấm ảnh để có thể đạt đến độ chính xác trên 95% (trên tập dữ liệu YouTube Faces). Việc tạo ra một tập dữ liệu khổng lồ ấy gần như là quá sức đối với bất kì một nhóm nghiên cứu phi lợi nhuận nào.

CNN đã gần như giải được bài toán hiệu năng cho các ứng dụng phân lớp nhưng lại yêu cầu một lượng dữ liệu huấn luyện khổng lồ để đạt được độ chính xác mong muốn. Điều gần như là không thể đối với những nhóm nghiên cứu nhỏ và những cộng đồng nghiên cứu phi lợi nhuận. Một bài toán mới lại được đặt ra với giới nghiên cứu, đó là phát triển nên một hệ thống phân lớp mới có độ chính xác cao nhưng yêu cầu một tập dữ liệu huấn luyện không quá lớn.

Không chỉ CNN, các thuật toán phân lớp trước đây cũng đều yêu cầu một tập dữ liệu đủ lớn để có thể đạt được độ chính xác mong muốn. Vì vậy, ý tưởng tạo ra một hệ thống phân lớp với độ chính xác cao nhưng chỉ cần ít dữ liệu nghe thật điên rồ. Nhưng hướng nghiên cứu mới, đã hiện thực hóa được vấn đề trên, đó chính là One-shot learning.

* 1. One-shot learning

Thông thường, một bài toán phân lớp đòi hỏi một tập dữ liệu đủ lớn gồm từ hàng nghìn đến hàng triệu mẫu dữ liệu tương ứng với mỗi lớp đối tượng cần phân loại để có thể huấn luyện được một mô hình phân lớp có độ chính xác tối ưu nhất.

One-shot learning phát triển những thuật toán không yêu cầu tập dữ liệu quá lớn như những thuật toán phân lớp trước đây, mỗi một lớp đối tượng cần phân loại chỉ cần một (hoặc một vài) mẫu dữ liệu để huấn luyện mô hình dùng để phân lớp.

One-shot learning được xây dựng thông qua việc quan sát quá trình học hỏi tri thức của con người, những “tri thức” cũ được chuyển đổi, trích xuất và sử dụng để hỗ trợ trong việc “học hỏi” những kiến thức mới. Bên cạnh đó, yếu tố quyết định đến thành công của One-shot learning chính là khả năng trích xuất thông tin tốt dựa vào những dữ liệu ít ỏi về đối tượng được cung cấp, điều này giúp cho quá trình huấn luyện đạt được hiệu quả tốt nhất. Ví dụ, một đứa trẻ có thể học cách nhận biết hươu cao cổ chỉ bằng một vài hình ảnh trong một cuốn sách hay cách những tín đồ Iphone có thể nhận biết mẫu Iphone mới dù chỉ xem qua một đoạn quảng cáo ngắn. Con người có thể dễ dàng làm được những việc này chỉ với một lượng ít ỏi dữ liệu trong khi những hệ thống deep learning tốt nhất cần đến hàng trăm hay thậm chí hàng nghìn mẫu dữ liệu để có được độ chính xác tối ưu nhất.

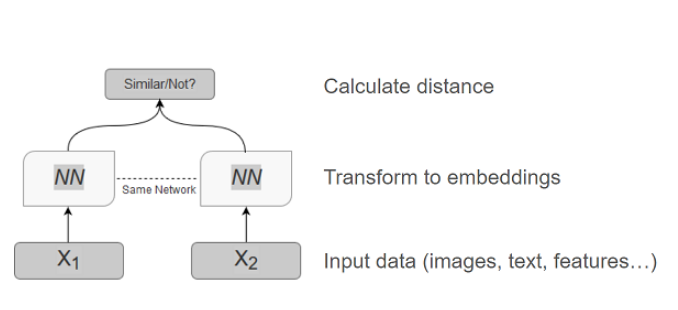
Như đã nói ở trên, CNN đã làm rất tốt với các bài toán phân lớp, đặc biệt là bài toán nhận diện khuôn mặt với một độ chính xác tương đối cao nhưng nó vẫn còn một điểm yếu là cần quá nhiều dữ liệu để đạt được độ chính xác ấy. One-shot learning ra đời với mục đích để khắc phục điểm yếu còn tồn đọng ấy của CNN. Nó thay đổi quan điểm của việc training từ trước đến nay là càng nhiều dữ liệu thì hiệu suất càng cao, thay vào đó là ý tưởng mới “ít nhưng chất”, One-shot learning chú trọng vào việc tăng khả năng trích xuất thông tin từ dữ liệu để gia tăng hiệu năng, bên cạnh đó, bản thân dữ liệu dùng để huấn luyện cũng phải chứa đựng nhiều thông tin hữu dụng nhất.

Việc huấn luyện một mô hình CNN với dữ liệu ít ỏi (với vài bức ảnh của đối tượng ở mỗi lớp) khiến bạn chỉ nhận được một mô hình phân lớp với hiệu năng cực kì kém. Bởi khả năng trích xuất thông tin kém cùng với dữ liệu dùng để huấn luyện không đủ lượng thông tin cần thiết để huấn luyện mô hình. Có rất nhiều giải pháp được đưa ra để giải quyết những vấn đề trên và Siamese network được xem là kiến trúc tối ưu nhất để ứng dụng one-shot learning vào CNN.

CHƯƠNG 2 – SIAMESE NETWORKS

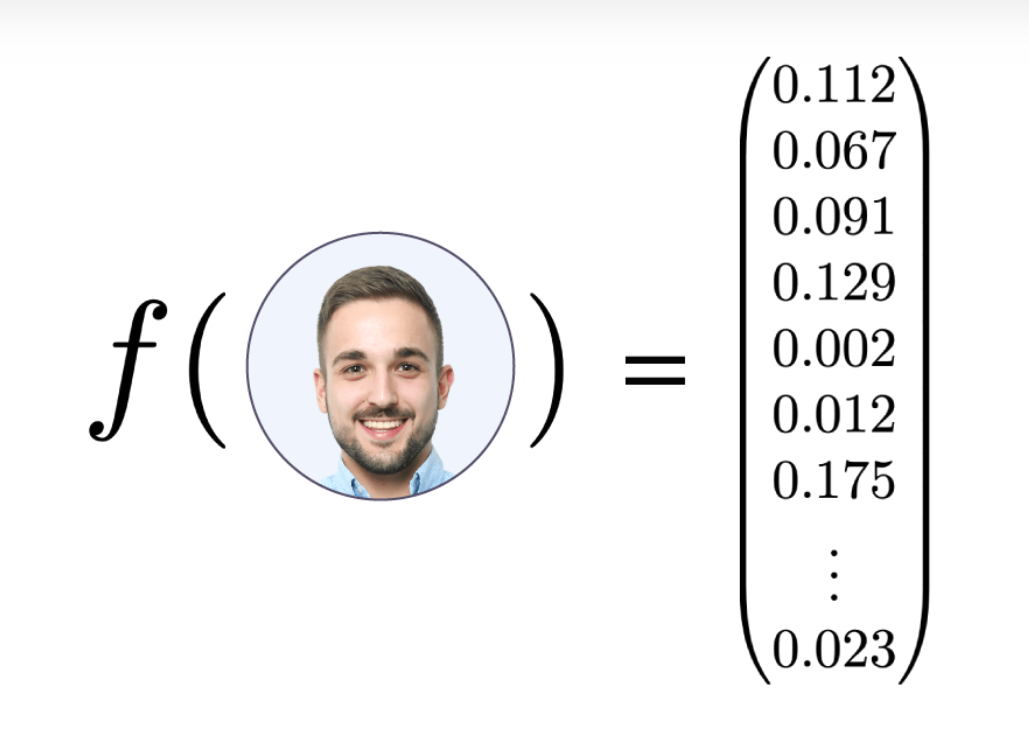
2.1 Siamese networks

2.1.1 Giới thiệu về Siamese network



Hình : Siamese networks đơn giản.

Siamese networks được xem là mô hình kiến trúc làm việc tốt nhất với One-shot learning. Nó được cấu tạo bởi hai mạng neural hoàn toàn giống nhau, mỗi mạng neural sẽ chuyển hóa một input đầu vào (là một ma trận điểm ảnh của một hình ảnh) thành một ma trận (n x 1). Ma trận này chính là biểu diễn cho các đặc trưng của bức ảnh (gọi là embedding). Lưu ý rằng hai hình ảnh sử dụng là hai hình ảnh khác nhau, có thể là cùng hoặc khác đối tượng mà chúng ta cần phân lớp.



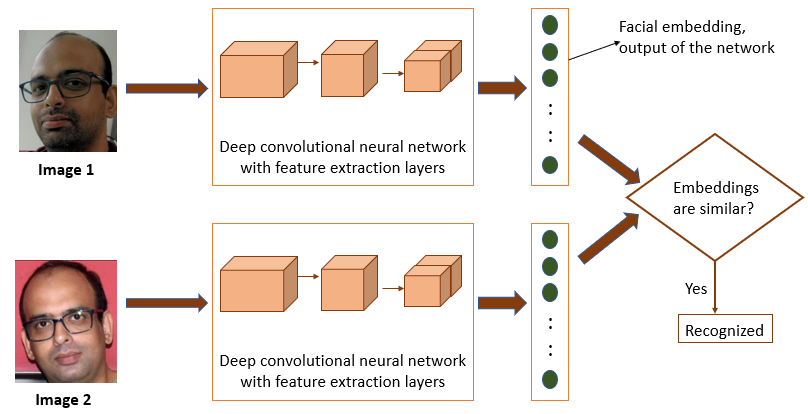
Hình : Ma trận đặc trưng (embedding).

Việc xác định hai hình ảnh này có phải là cùng lớp đối tượng hay không được thực hiện ở bước tiếp theo, bằng công thức tính khoảng cách Euclidean giữa hai hình ảnh:

Trong đó:

* x là ma trận đặc trưng (embedding) của hình ảnh thứ nhất.
* y là ma trận đặc trưng (embedding) của hình ảnh thứ hai.
* và là phần tử thứ i của hai ma trận.

Giá trị khoảng cách Euclidean giữa hai hình ảnh biểu diễn cho sự khác biệt giữa hai hình ảnh đó. Khoảng cách càng lớn càng chứng tỏ hai hình ảnh không thuộc cùng một đối tượng và ngược lại.



Hình : Quá trình xử lí dữ liệu của một mạng siamese network.

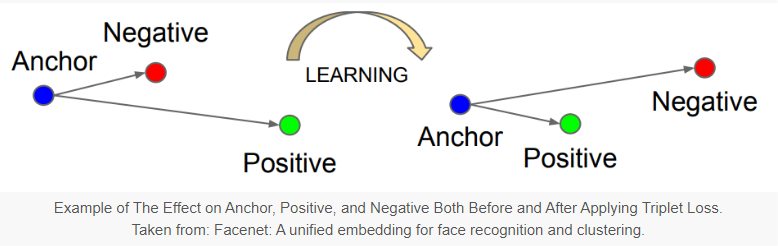
2.1.2 Triplet loss

Thông thường, quá trình học của Siamese networks được thực hiện nhờ triplet loss, đây là một phương pháp thay thế giúp tăng hiệu quả khi huấn luyện mô hình. Mô hình sẽ cần 3 hình ảnh cho mỗi lượt huấn luyện và độ mất mát sẽ được tính như sau:

* + Chuyển ảnh đầu vào thành vec-tơ dữ liệu thông qua neural network, ta thu được ma trận đặc trưng, gọi là Anchor input (A).
  + Chuyển ảnh cùng lớp đối tượng với ảnh Anchor thành vector dữ liệu thông qua neural network, ta thu được ma trận đặc trưng, gọi là Positive input (P).
  + Chuyển ảnh khác lớp đối tượng với ảnh Anchor thành vector dữ liệu thông qua neural netwwork, ta thu được ma trận đặc trưng, gọi là Negative input (N).
  + Tính khoảng cách Euclidean d(A,P), d(A,N). Ý tưởng của việc huấn luyện chính là tiến hành cực tiểu d(A, P) và cực đại d(A,N).
  + Hàm mất mát được định nghĩa như sau:

L = max(d(a, p)- d(a,n) + α, 0),

Trong đó, α là tham số ngưỡng giúp phân biệt hai hình hình ảnh là cùng một lớp đối tượng (positive) hay khác lớp đối tượng (negative). Hai hình ảnh được xem là thuộc cùng một lớp (positive) khi khoảng cách Euclidean của chúng bé hơn α và ngược lại.

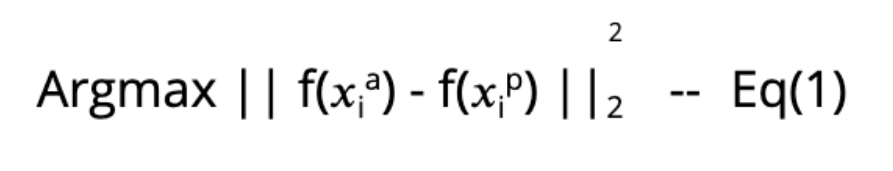


Hình : Triplet loss.

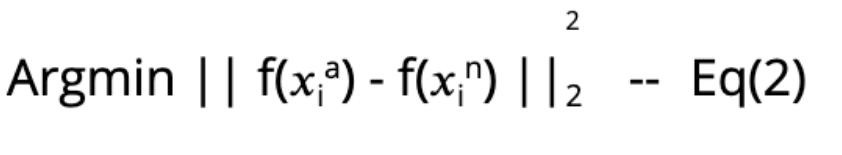
2.1.2 Triplet Selection

Việc lựa chọn các cặp hình ảnh để huấn luyện cũng cực kì quan trọng. Nếu những cặp ảnh được lựa chọn quá phù hợp với những điều kiện đã nêu ở trên, tức là mô hình của bạn sẽ chẳng thể cải thiện được gì lúc này, điều này khiến tốc độ học của mô hình sẽ bị giảm đáng kể. Để đảm bảo tốc độ huấn luyện của mô hình không bị làm chậm đi, chúng ta cần phải cố gắng lựa chọn những cặp hình ảnh càng vi phạm với các ràng buộc triplet càng tốt. Nguyên tắc chọn các cặp hình ảnh để huấn luyện như sau:

* Nguyên tắc chọn cặp ảnh anchor-positive là tìm những cặp ảnh có độ khác biệt lớn nhất. Được mô tả bằng công thức Eq(1) dưới đây.



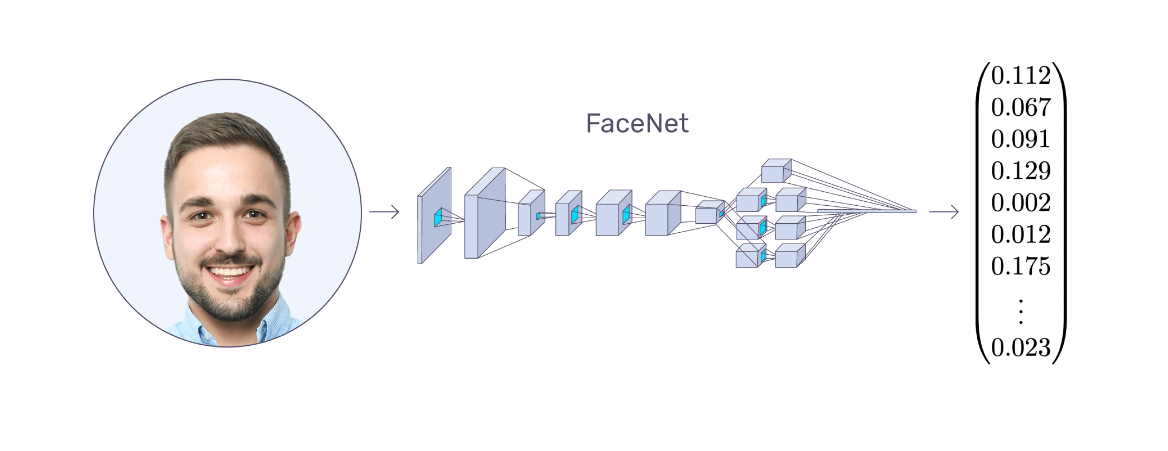
* Nguyên tắc chọn cặp ảnh anchor-negative là tìm những cặp ảnh có độ khác biệt nhỏ nhất. Được mô tả bàng công thức Eq(2) bên dưới.



Những cặp ảnh anchor-positive thỏa mản Eq(1) được gọi là hard positive. Còn những cặp ảnh anchor-negative thỏa mản Eq(2) được gọi là hard negative. Đây là những cặp dữ liệu huấn luyện có lượng thông tin cao giúp đẩy nhanh quá trình huấn luyện của mô hình. Tuy nhiên, việc áp dụng triplet selection có một vấn đề phát sinh khi phải chọn ra những cặp hình ảnh phục vụ việc huấn luyện trên trên toàn bộ dữ liệu huấn luyện. Việc tính toán để lựa chọn trên toàn bộ tập dữ liệu là gần như không thể hoặc vô cùng tốn kém tài nguyên, ảnh hưởng đến quá trình huấn luyện mô hình. Vì vậy, thay vì phải chọn những cặp hard positive và hard negative trên toàn bộ tập dữ liệu, chúng ta chỉ cần chọn trên tập con của tập dữ liệu ở mỗi lượt huấn luyện, tập con này có kích cỡ vừa phải, để đảm bảo chọn được những cặp ảnh phù hợp mà không ảnh hưởng đến quá trình huấn luyện.

2.2 FaceNet

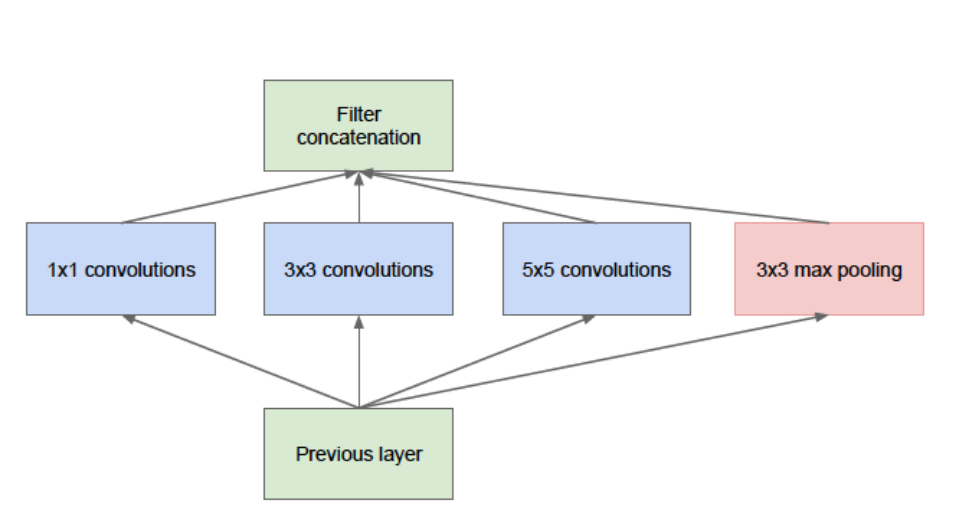
FaceNet là kiến trúc mạng CNN được phát triển bởi Google vào năm 2015 dựa trên kiến trúc mạng sinh đôi (siamese network). Nó chính là thành quả của quá trình nghiên cứu, vận dụng One-shot learning vào quá trình huấn luyện. Không giống với các kiến trúc CNN khác, FaceNet không có lớp cổ chai (bottleneck layer) với nhiệm vụ phân lớp đối tượng. Thay vào đó, FaceNet chỉ ánh xạ một ma trận điểm ảnh của một hình ảnh đầu vào thành một ma trận đặc trưng (embedding vector). Từ những ma trận đặc trưng này, việc phân lớp đối tượng có thể được thực hiện bằng các công nghệ phân lớp khác, ví dụ như k-NN (phân cụm K-nearest neighbor). Nhóm nghiên cứu của Google đã thử nghiệm rất nhiều kiến trúc CNN khác nhau cho FaceNet, và Inception được xem là kiến trúc phù hợp nhất.



Hình : FaceNet biến đổi hình ảnh đầu vào thành ma trận đặc trưng.

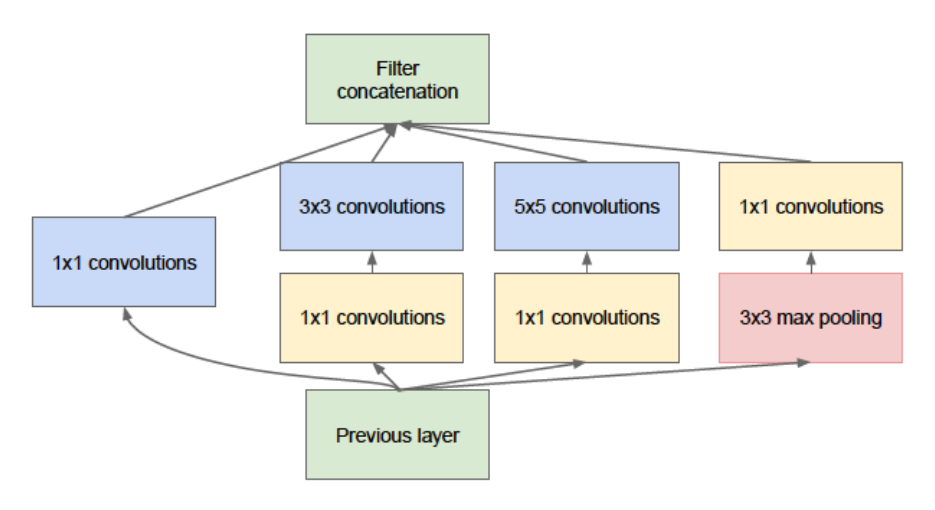
2.2.1 Kiến trúc Inception

Ý tưởng chính của kiến trúc Inception là sử dụng đồng thời nhiều bộ lọc có kích thước khác nhau. Trong những kiến trúc CNN truyền thống khác, mỗi lớp thông thường chỉ có một bộ lọc duy nhất được sử dụng. Nhưng trong kiến trúc Inception, chúng ta có thể sử dụng nhiều bộ lọc khác nhau trong mỗi lớp.



Hình : Inception module.

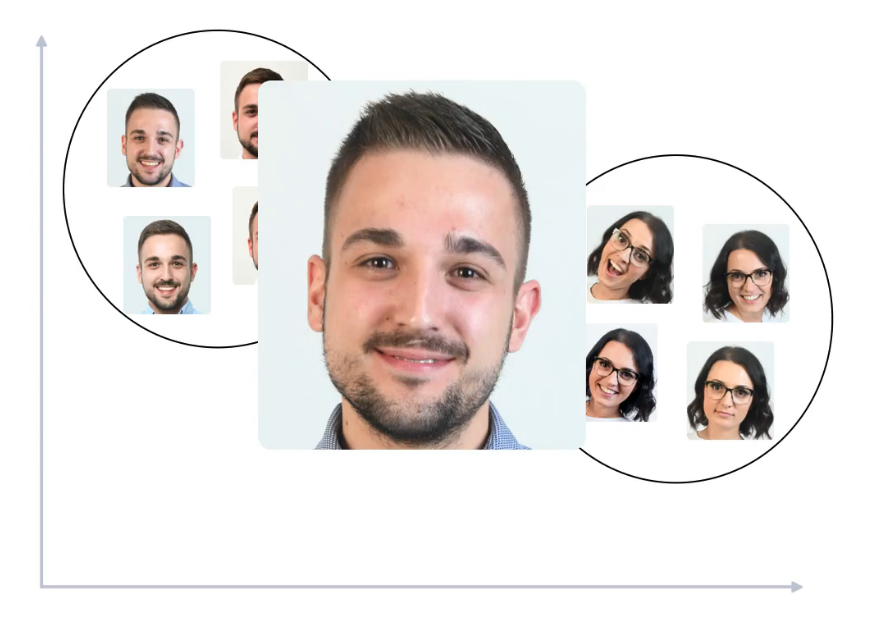
Việc sử dụng nhiều bộ lọc đồng thời sẽ khiến quá trình tính toán nặng nề, để giảm tải quá trình tính toán, các nhà nghiên cứu đã sử dụng thêm các ma trận giảm chiều theo sau các bộ lọc để giảm chiều các ma trận, từ đó khiến việc tính toán trở nên dễ dàng hơn.



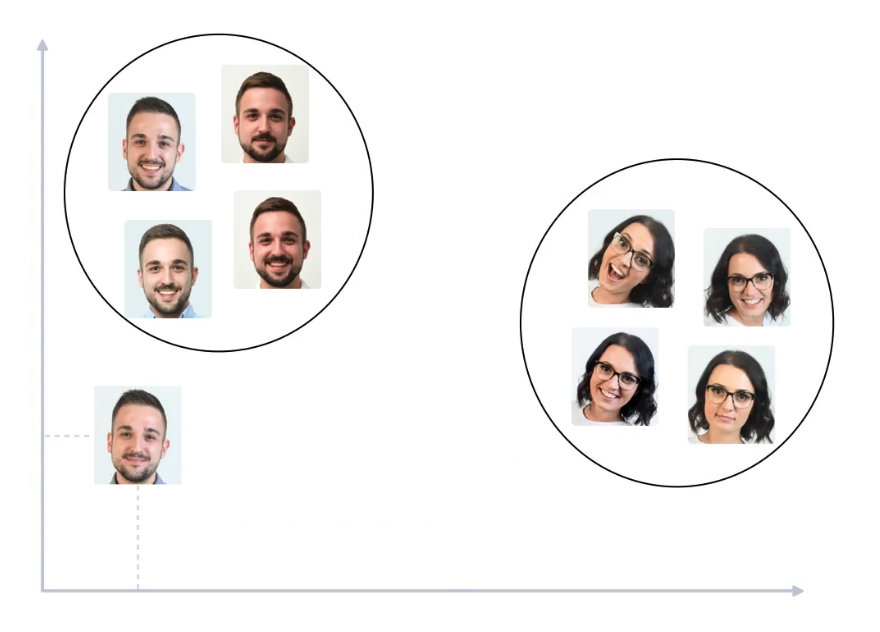
Hình : Inception module sử dụng các ma trận giảm chiều.

2.2.3 Cách hệ thống nhận diện một đối tượng.

* Ta gọi bức ảnh cần phân loại là X. Sau đó ta chuyển đổi ma trận điểm ảnh của X thành ma trận đặc trưng (embedding) thông qua mạng Inception.

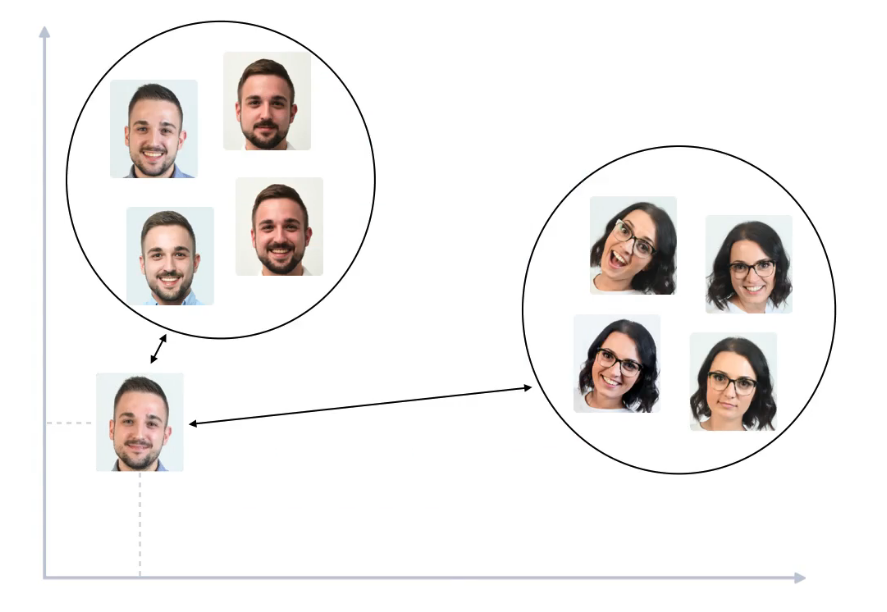


Hình : Bức ảnh cần phân loại.



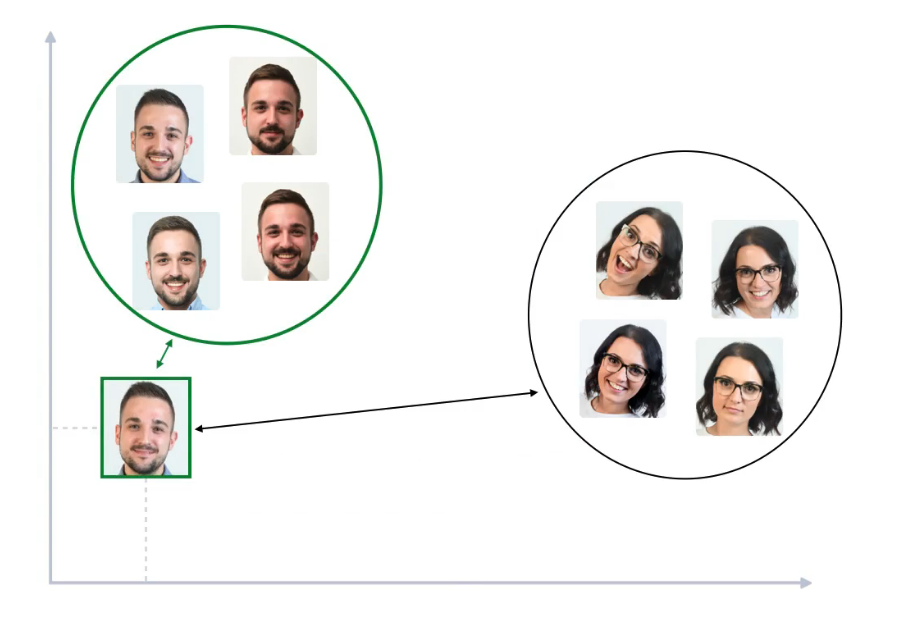
Hình : Chuyển bức ảnh cần phân loại thành ma trận đặc trưng.

* Sử dụng khoảng cách Euclidean để lần lượt tính khoảng cách giữa ma trận đặc trưng (embedding) của X với các ma trận đặc trưng của các hình ảnh trong tập dữ liệu.



Hình : Tính khoảng cách giữa X và các hình ảnh trong tập dữ liệu.

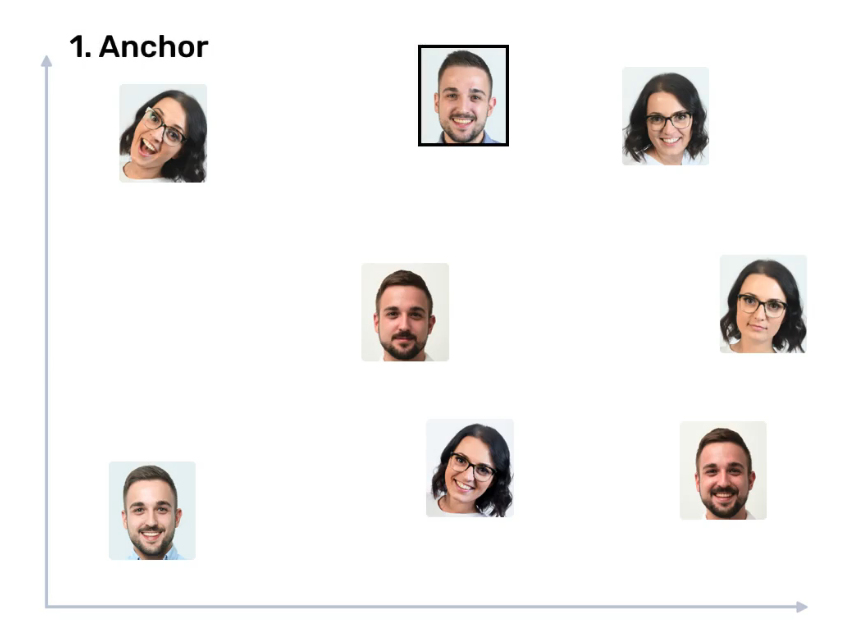
* Sau đó so sánh với các khoảng cách trên với nhau, lựa chọn khoảng cách nhỏ nhất, X được xem là cùng lớp đối tượng với bức ảnh có khoảng cách Euclidean nhỏ nhất với nó.



Hình : Lựa chọn lớp đối tượng mà X thuộc về dựa vào khoảng cách Euclidean.

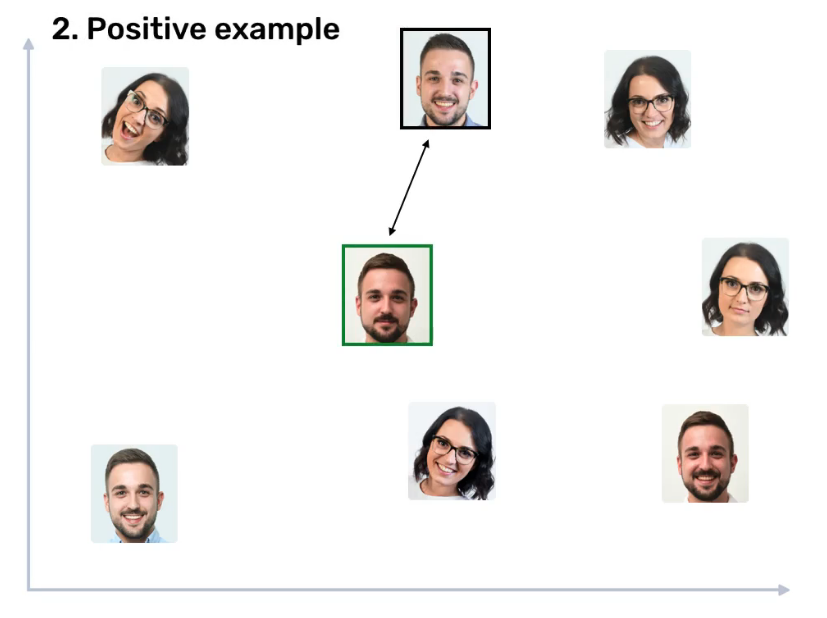
2.2.2 Quá trình huấn luyện FaceNet

* Bước 1: Chọn một embedding anchor ngẫu nhiên.



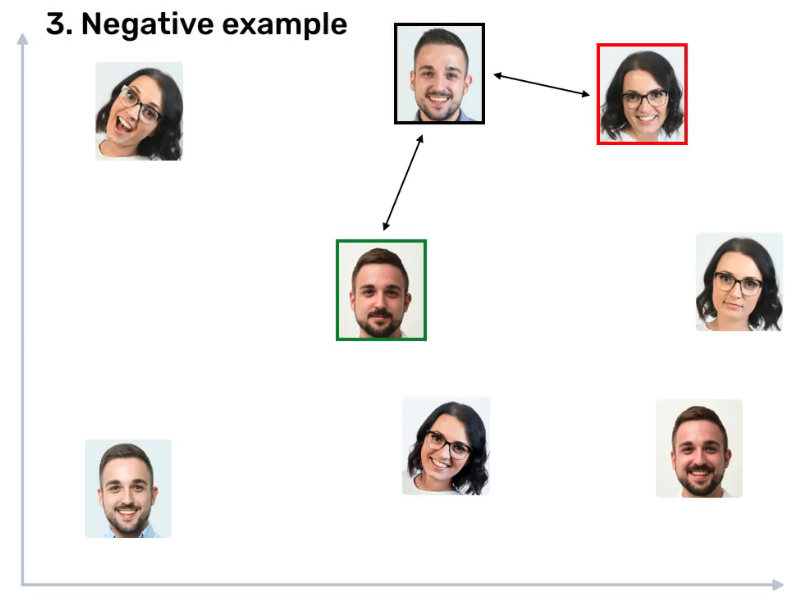
Hình : Chọn Anchor.

* Bước 2: Chọn embedding positive dựa vào anchor đã chọn trước đó.



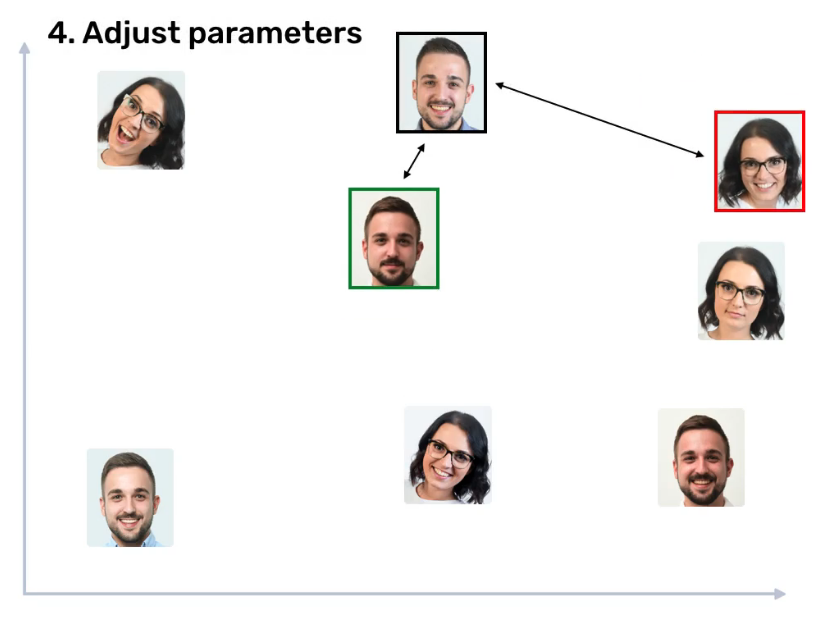
Hình : Chọn Positive Image.

* Bước 3: Chọn embedding negative dựa vào anchor đã chọn trước đó.



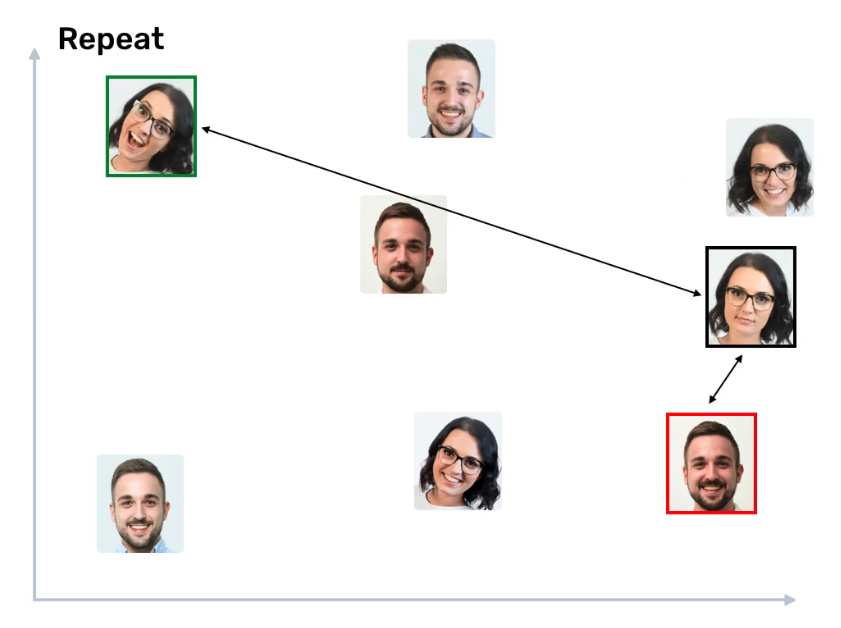
Hình : Chọn Negative image.

* Bước 4: Điều chỉnh các trọng số của mô hình theo ràng buộc của triplet, sao cho d(a,p) càng nhỏ càng tốt, và d(a,n) càng lớn càng tốt.

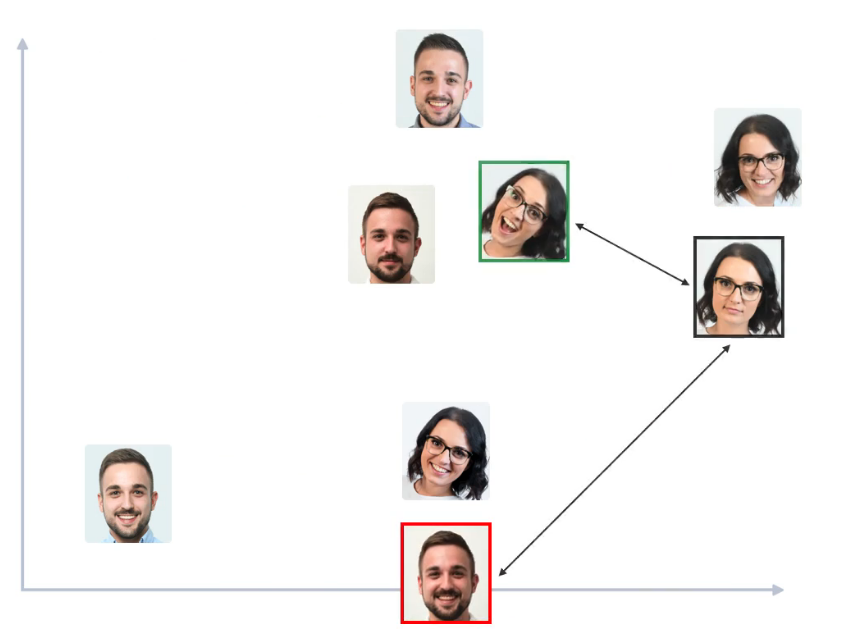


Hình : Điều chỉnh các trọng số (weight).

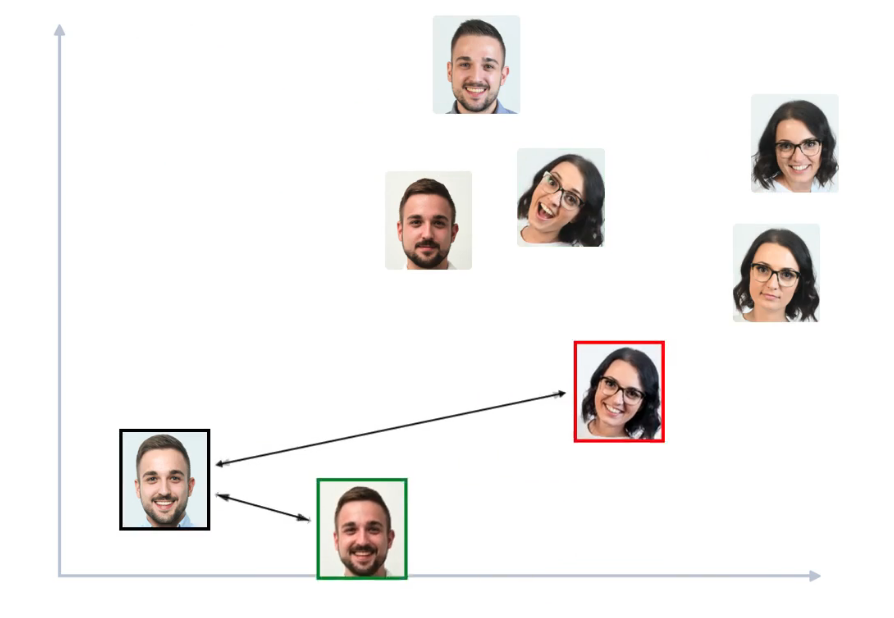
* Bước 5: Lặp lại các bước trên.



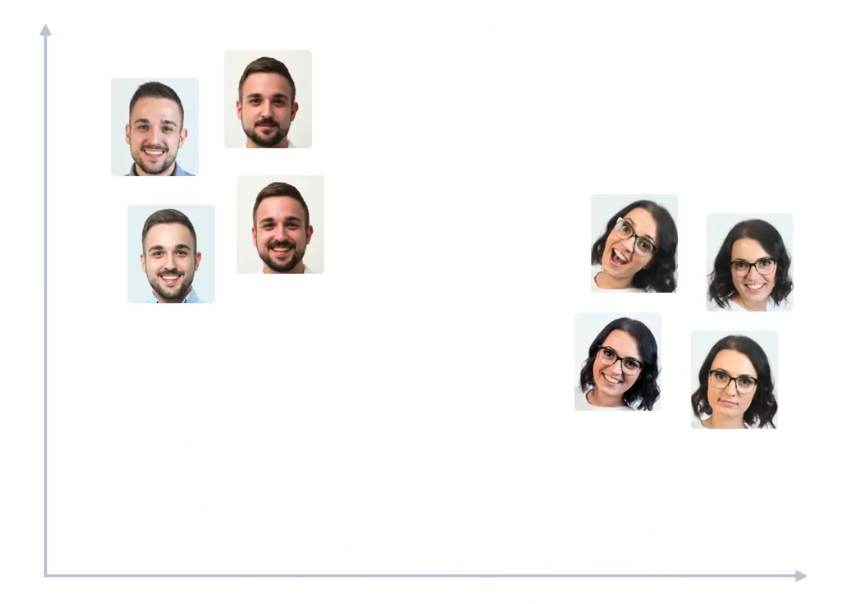
Hình : Lặp lại các bước trên với một anchor khác.



Hình : Lặp lại với một anchor khác.



Hình : Tiếp tục lặp lại bước trên với một anchor khác.



Hình : Kết quả.

CHƯƠNG 3 – XÂY DỰNG ỨNG DỤNG NHẬN DẠNG KHUÔN MẶT

3.1 Một số khái niệm cần phân biệt

3.1.1 Điểm khác biệt giữa face verification và face recognition.

Công nghệ face verificationđược sử dụng để xác minh tính hợp lệ của một hình ảnh đầu vào so với một hình ảnh chuẩn cho trước (ánh xạ 1 đến 1). Mặt khác, face recognition được sử dụng để nhận diện xem một ảnh đầu vào có thuộc một nhóm hình ảnh cho trước hay không (ánh xạ 1 đến M).

3.2.2 Điểm khác biệt giữa nhận diện khuôn mặt và phát hiện giả mạo khuôn mặt.

Như đã thể hiện trong phần chụp màn hình của ứng dụng, ứng dụng chỉ có thể nhận diện khuôn mặt cơ bản , có thể nhận diện khuôn mặt từ mọi nguồn không phân biệt giữa hình ảnh kỹ thuật số 2D hay khuôn mặt mô hình 3D.

Ngăn chặn việc sử dụng các khuôn mặt “không sống” cho việc qua mặt những hệ thống face recognition được gọi là face spoofing detection (phát hiện giả mạo khuôn mặt), việc này nằm ngoài phạm vi của ứng dụng này. Phát hiện giả mạo khuôn mặt có thể đạt được bằng cách kết hợp các kỹ thuật đa dạng chẳng hạn như liveness detection, contextual information, user interaction, và texture analysis.

Huấn luyện một convolution neural network (CNN) mới để nhận diện khuôn mặt là cực kì khó khăn vì sự phức tạp về độ lớn của bộ dữ liệu và yêu cầu rất lớn của khả năng tính toán của hệ thống. Do đó, tôi sẽ tạo ra một ứng dụng nhận biết khuôn mặt bằng việc sử dụng một mô hình được huấn luyện từ trước trong những dự án open source.

**TÀI LIỆU THAM KHẢO**

**Tiếng Anh**