**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC DUY TÂN**



**Đề tài:**

[**XÂY DỰNG**](http://www.ktcn.tvu.edu.vn/images/stories/phieudexuat_pdf/KL12_04_Le%20Thi%20Thuy%20Lan.pdf) **WEBSITE DẠY VÀ HỌC TIẾNG ANH TÍCH HỢP NHẬN DIỆN KHUÔN MẶT**

🙠🖎🕮✍🙢

**TÀI LIỆU** **THIẾT KẾ THUẬT TOÁN**

***GVHD:*** THS. NGUYỄN TẤN THUẬN

***Nhóm sinh viên*:**

Nguyễn Tấn Lộc– 2321117975

Huỳnh Đình Quốc – 2321118144

Trịnh Minh Hậu – 23211410671

Huỳnh Như Ngọc – 2320121339

Lê Thị Bảo Quyên – 2320118093

**Đà Nẵng, tháng 3 năm 2021**

MỤC LỤC

[GIỚI THIỆU 2](#_Toc73199204)

[**1.1.** **Đặt vấn đề** 2](#_Toc73199205)

[**1.2. Ngôn ngữ Python** 2](#_Toc73199206)

[**1.3. Công nghệ AI nhận diện khuôn mặt** 5](#_Toc73199207)

[1.3.1. Khái niệm 5](#_Toc73199208)

[1.3.2. Các bước nhận diện khuôn mặt(*Face Recognition)* 7](#_Toc73199209)

[1.3.3. Một số thuật toán nhận diện khuôn mặt 10](#_Toc73199210)

[1.3.4. Thuật toán MTCNN và Facenet của dự án 17](#_Toc73199211)

# GIỚI THIỆU

* 1. **Đặt vấn đề**

Năm 2019, công nghệ AI có những bước tiến thần tốc tại thị trường Việt Nam, len lỏi ở mọi mặt của đời sống, không chỉ riêng trong ngành công nghệ. Ứng dụng công nghệ này, điểm danh bằng nhận diện khuôn mặt FaceID đã được phát triển và triển khai ở các trường học. Hình điểm danh này linh hoạt theo từng cá nhân. Mỗi học sinh/ sinh viên có thể tự theo dõi số ngày đi học, thời gian đi muộn,… Bên cạnh đó, chấm công FaceID cũng đơn giản hóa quy trình điểm danh. Cùng với sự phát triển đó, các trường học cũng không ngừng phát triển khi áp dụng những công nghệ mới trong lĩnh vực này.

Về bản chất, điểm danh bằng nhận diện khuôn mặt là hoạt động sinh viên báo cáo với giảng viên về việc tham gia lớp học cho đến khi kết thúc môn đó trong một ngày. Việc điểm danh này sẽ giúp cho nhà trường và giảng viên nắm được số sinh viên tham gia lớp học đó, dẫn đến việc điểm danh sẽ trở nên dễ dàng và đảm bảo hơn về mặt chính xác. Nếu trước đây, các trường học thực hiện công việc điểm danh các sinh viên bằng cách thủ công qua ghi chép sổ sách hay qua excel thì hiện nay có một số trường đại học lớn đã áp dụng công nghệ để thực hiện hoạt động này. Do vậy, nhu cầu thiết kế ứng dụng điểm danh bằng nhận diện khuôn mặt cho các trường học là rất cần thiết. Vấn đề đặt ra là ứng dụng phải đảm báo việc xử lý dữ liệu nhanh và tính chính xác cao.

* 1. **Ngôn ngữ Python**

Python có tính hiệu quả rất cao với các cấu trúc dữ liệu và đơn giản nhưng lại rất hữu dụng với ngôn ngữ lập trình hướng đối tượng. Cú pháp Python gọn gàng và có tính năng gõ động; cùng với tính diễn dịch tự nhiên, làm Python là 1 ngôn ngữ lý tưởng dành cho viết kịch bản và phát triển ứng dụng nhanh chóng trong nhiều lĩnh vực trên nhiều nền tảng khác nhau.

Guido van Rossum, người tạo ra ngôn ngữ Python, cho biết tên Python (có nghĩa là rắn) xuất phát từ việc ông không thích con rắn ăn bằng cách cuốn chặt con mồi.

1. **Lợi ích khi sử dụng ngôn ngữ Python**

Đơn giản: Python là 1 ngôn ngữ đơn giản và tối giản. Đọc 1 chương trình Python có cảm giác như đọc tiếng Anh, mặc dù ở dạng rút gọn. Tính tự nhiên của mã giả trong Python là 1 trong các điểm mạnh nhất của ngôn ngữ này. Điều này giúp cho lập trình viên tập trung vào giải pháp giải quyết vấn đề hơn là việc tập trung vào ngôn ngữ.

Dễ học: Python dễ học vì có cú pháp cực kỳ đơn giản.

Miễn phí và mã nguồn mở: Python là 1 ví dụ của FLOSS (Free/Libré and Open Source Software). Vì vậy, bạn có thể tự do phân phối bản sao chép của phần mềm, cũng như mã nguồn, thay đổi hay sử dụng các thành phần phần mềm trong các chương trình mới. Một trong những lý do Python là ngôn ngữ mạnh vì nó được cộng đồng thường xuyên phát triển và nâng cấp.

Ngôn ngữ bậc cao: Khi sử dụng Python, bạn sẽ không bao giờ phải để ý đến các chi tiết mức thấp như quản lý bộ nhớ cho chương trình của bạn, và vân vân.

Khả năng bỏ túi: Do tính tự nhiên mã mở của Python, Python cũng xây dựng chạy trên nhiều nền tảng khác nhau. Bạn có thể sử dụng Python trên GNU/Linux, Windows, FreeBSD, Macintosh, Solaris, OS/2, Amiga, AROS, AS/400, BeOS, OS/390, z/OS, Palm OS, QNX, VMS, Psion, Acorn RISC OS, VxWorks, PlayStation, Sharp Zaurus, Windows CE và PocketPC. Bạn còn có thể dùng một nền tảng như Kivy để tạo các trò chơi trên máy tính dành cho iPhone, iPad, và Android.

Diễn dịch: Khi 1 chương trình được viết bằng ngôn ngữ biên dịch (như C hoặc C++) thì nó được chuyển đổi từ mã ngôn ngữ (C/C++) thành ngôn ngữ mà máy tính có hiểu được bằng cách dùng 1 trình biên dịch với các chức năng và các cờ khác nhau. Trái lại, Python không cần biên dịch ra nhị phân. Chương trình viết bằng Python chạy trực tiếp từ mã nguồn. Cụ thể, Python sẽ chuyển mã nguồn thành 1 dạng trung gian gọi là bytecode, sau đó dịch dạng trung gian thành ngôn ngữ mà máy tính có thể hiểu được.

Hướng đối tượng: Python là ngôn ngữ hỗ trợ cho lập trình hướng đối tượng lẫn cả lập trình thủ tục. Nếu so sánh với C++ hoặc Java, Python rất mạnh nhưng lại cực kỳ đơn giản để thực hiện lập trình hướng đối tượng.

Tính mở rộng: Nếu bạn cần 1 đoạn mã chạy nhanh hoặc 1 vài thuật toán đóng, bạn có thể lập trình ở C/C++ và sau đó sử dụng nó cho chương trình Python. Python cho phép tích hợp các chương trình ở các ngôn ngữ khác.

Khả năng nhúng: Bạn có thể nhúng Python bên trong chương trình C/C++ để mang lại các khả năng kịch bản cho người dùng chương trình.

1. **Thư viện mở rộng:**

Thư viện tiêu chuẩn Python thì rất lớn. Thư viện giúp bạn làm nhiều thứ khác nhau liên quan đến biểu thực chính quy, gieo tài liệu, tiến trình/tiểu trình, database, trình duyệt web, CGI, FTP, email, XML, XML-RPC, HTML, tập tin WAV, mã hóa, GUI, và các phần khác. Tất cả thứ này đều sẵn có khi cài đặt Python. Trong đó thư viện PyQt5 hỗ trợ cho việc xử lý ảnh. PyQt5 là một thư viện công cụ Python cho khung QT5 . Nó có sẵn cho Python 3.x. PyQt5 được triển khai như một tập hợp các mô-đun Python. Nó cho phép chúng tôi liên kết các thiết kế chúng tôi thực hiện với QT Designer và sau đó có thể dịch chúng với Py Qt để chúng có thể được thực thi theo Python trên tất cả các hệ điều hành như Linux Windows và Mac OS. Để có thể phát triển ứng dụng, chúng ta phải cài đặt các chương trình và phụ thuộc cần thiết, hướng dẫn này sẽ sử dụng phiên bản Linux Mint làm hệ điều hành nhưng có thể được điều chỉnh cho mọi hệ điều hành. Trong các hướng dẫn khác, chúng tôi sử dụng Qt để lập trình trong C ++, Lập trình đa nền tảng trong C ++ và Qt trong cơ sở dữ liệu GNU và Môi trường Linux và Lập trình QT và SQLite.

Các class của PyQt5 được chia thành các module, bao gồm :

+ QtCore : là module bao gồm phần lõi không thuộc chức năng GUI, ví dụ dùng để làm việc với thời gian, file và thư mục, các loại dữ liệu, streams, URLs, mime type, threads hoặc processes.

+ QtGui : bao gồm các class dùng cho việc lập trình giao diện (windowing system integration), event handling, 2D graphics, basic imaging, fonts và text.

+ QtWidgets : bao gồm các class cho widget, ví dụ : button, hộp thoại, … được sử dụng để tạo nên giao diện người dùng cơ bản nhất.

+ QtMultimedia : thư viện cho việc sử dụng âm thanh, hình ảnh, camera,…

+ QtBluetooth : bao gồm các class giúp tìm kiếm và kết nối với các thiết bị có giao tiếp với phần mềm.

+ QtNetwork : bao gồm các class dùng cho việc lập trình mạng, hỗ trợ lập trình TCP/IP và UDP client , server hỗ trợ việc lập trình mạng.

+ QtPositioning : bao gồm các class giúp việc hỗ trợ xác định vị.

+ Enginio : module giúp các client truy cập các Cloud Services của Qt.

+ QtWebSockets : cung cấp các công cụ cho WebSocket protocol.

+ QtWebKit : cung cấp các class dùng cho làm việc với các trình duyệt Web , dựa trên thư viện WebKit2.

+ QtWebKitWidgets : các widget cho WebKit.

+ QtXml : các class dùng cho làm việc với XML file.

+ QtSvg : dùng cho hiển thị các thành phần của SVG file.

+ QtSql : cung cấp các class dùng cho việc làm việc với dữ liệu.

+ QtTest : cung cấp các công cụ cho phép test các đơn vị của ứng dụng với PyQt5.

* 1. **Công nghệ AI nhận diện khuôn mặt**
     1. **Khái niệm**

công nghệ AI nhận dạng khuôn mặt là công nghệ cho phép nhận dạng một người cụ thể từ ảnh hoặc 1 đoạn video. Cách thức làm việc của công nghệ này là so sánh hình ảnh khuôn mặt với những hình ảnh sẵn có trong cơ sở dữ liệu để đưa ra kết quả.

Hệ thống này thường được sử dụng trong các hệ thống an ninh và có thể được so sánh với các dạng sinh trắc học khác như các hệ thống nhận dạng vân tay hay tròng mắt.

* + 1. **Cách thức hoạt động**

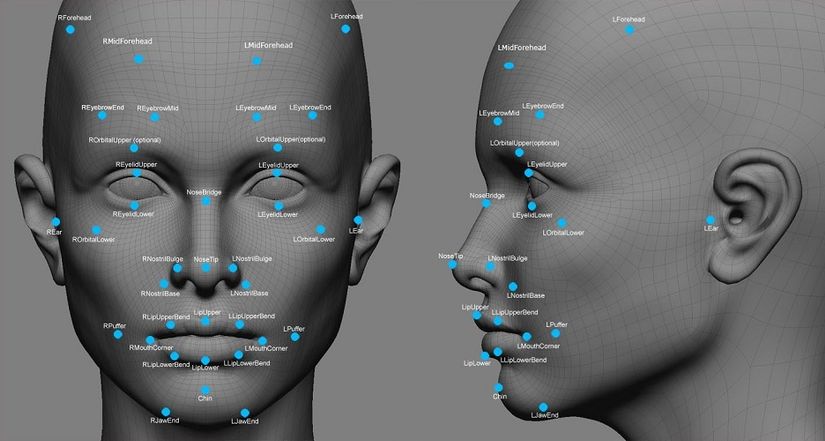
Quá trình mà sinh trắc học nhận diện khuôn mặt hoạt động bao gồm:

1. Lấy mẫu: Trước hết, để có thể phân tích khuôn mặt và nhận diện, cần phải tách khuôn mặt ra khỏi khung cảnh còn lại trước đã
2. Phân tích: Chúng ta nhận diện gương mặt mọi người qua các đặc điểm, hệ thống nhận diện cũng vậy, nhưng nó thực hiện đánh giá các đặc điểm ở 1 level cao hơn
3. So sánh: Mỗi khuôn mặt đều có nhiều điểm mốc, những phần lồi lõm tạo nên các đặc điểm của khuôn mặt. Các hệ thống nhận diện gương mặt định nghĩa những điểm này là những điểm nút:
4. Kết quả: Hệ thống sau đó sẽ quyết định kết quả so sánh có phù hợp hay không.

* Khi đối tượng lọt vào tầm quét của camera, hệ thống sẽ tự động xác định vị trí khuôn mặt và thực hiện các thuật toán để trích xuất các dữ liệu trên khuôn mặt. Trong quá trình di chuyển, đối tượng sẽ giúp hệ thống dễ dàng xác nhận khuôn mặt hơn. Tất cả các quá trình trên diễn ra trong vòng chưa đầy 2 giây.
  + 1. **Đặc điểm nổi bật**
* Công nghệ nhận diện khuôn mặt có nhiều ưu điểm mà các phương pháp xác thực khác không có
* Một trong những khía cạnh tích cực của nhận diện khuôn mặt là hệ thống không cần sự tương tác của con người dùng,
* Không đòi hỏi người dùng phải chờ đợi trong thời gian dài hoặc làm bất cứ điều gì hơn là nhìn vào camera.
* Kiểm soát theo thời gian thực (real-time).
  + 1. **Nhận dạng 3D**

Ngày nay, với sự phát triển của công nghệ, và với xu hướng 3D đang thịnh hành, thì cũng tương tự vậy, chúng ta có nhận dạng khuôn mặt 3 chiều. Kỹ thuật này sử dụng các cảm biến 3D để nắm bắt thông tin về hình dạng của khuôn mặt, rồi dùng các điểm nổi bật trên khuôn mặt – nơi những mô cứng và xương nhìn thấy rõ nhất như đường cong của hốc mắt, mũi và cằm - để nhận ra đối tượng. Các đặc điểm này là độc nhất đối với mỗi khuôn mặt và không thay đổi theo thời gian.

Cách thức sử dụng độ sâu và trục của các phần trên khuôn mặt không bị ảnh hưởng bởi ánh sáng, vì thế việc nhận dạng khuôn mặt 3D có thể được sử dụng cả trong bóng tối và có thể nhận ra khuôn mặt từ nhiều góc độ khác nhau với độ chênh lệch lên tới 90 độ. Và nhờ có việc xác định khuôn mặt dựa trên hình ảnh 3 chiều, nên nhận dạng 3D có khả năng xác định 1 khuôn mặt từ nhiều góc nhìn hơn. Số lượng các điểm dữ liệu 3 chiều khiến cho độ chính xác tăng lên đáng kể bởi sự phát triển của các bộ cảm biến tinh vi giúp nắm bắt hình ảnh chụp khuôn mặt 3D được tốt hơn.



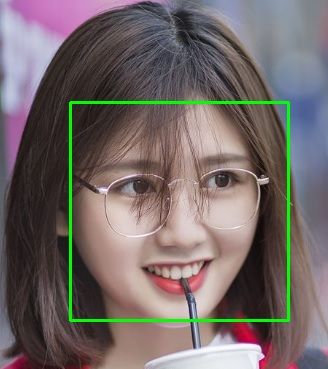
* 1. **Các bước nhận diện khuôn mặt(*Face Recognition)***

Tuỳ theo cách tiếp cận khác nhau nhưng về cơ bản một bài toán face recognition sẽ gồm các bước sau: Face Detection, Face Alignment, Face Respresentation, Face Recognization. Đây cũng chính là các bài toán con mà chúng ta cần phải giải quyết, mỗi bài toán con sẽ có cách tiếp cận, cách giải và độ khó khác nhau, vì vậy độ chính xác ở các bài toán(bước) tiếp theo sẽ phụ thuộc rất nhiều vào độ chính xác của các bài toán(bước) trước đó. Hơn nữa, để có thể vừa đạt được độ chính xác cao cũng như vừa chạy được realtime là một điều không phải dễ dàng.

* + 1. **Face Detection**

Đây là bước cực kỳ quan trọng để đảm bảo các bước sau của bạn có đạt được độ chính xác hay không. Ở bước này, chúng ta sẽ cần một mô hình có khả năng nhận biết ở đâu trong một bức ảnh có sự xuất hiện của mặt người, mô hình sẽ trả về cho chúng ta các toạ độ (x1, y1, x2, y2) hoặc (x, y, w, h) để hợp thành các ô vuông là các boundingbox (ô xanh trên hình là bounding box), các boundingbox này sẽ chứa mặt người bên trong từ đó giúp chúng ta thực hiện crop face, trích xuất ra đặc trưng là mặt của bức ảnh. Ngoài ra, mô hình cũng sẽ trả về cho chúng ta các tham số như: confidence(độ tự tin vị trí này có mặt người), và các toạ độ trên khuôn mặt như vị trí của mắt, mũi, cằm, lông mày, miệng,...

Để giải quyết bài toán này, các bạn có thể dùng các thư viện có sẵn như **MTCNN**, **Dlib**, **Dnn** của **OpenCV**. Hoặc nếu không các bạn có thể tự build cho mình một mô hình CNN và train lại từ đầu, của mình thì mình dựa trên MobileNet SSD thu được kết quả tương đối tốt cả về độ chính xác và thời gian predict của mô hình.



* + 1. **Face Alignment**

Đúng như tên gọi của nó alignment (sự căn chỉnh), sau bước detection các faces của chúng ta có thể ở các trạng thái khác nhau, các góc độ khác nhau, có những faces bị chéo và cũng có thể bị lệch do bước detect chưa chính xác trong việc lấy ra khung hình chuẩn của mặt. Thì việc áp dụng face alignment ở đây là cần thiết, nó có thể hiểu như một hình thức của data normalization, giúp tiêu chuẩn hoá lại dữ liệu trước khi đưa vào mô hình dự đoán identity.

Sau khi crop được face từ bước detect, căn chỉnh lại face từ bước alignment, việc tiếp theo chúng ta cần làm là biểu diễn các face này thành các vector đặc trưng trước khi đưa vào mô hình dự đoán indentity. Có thể hiểu đơn giản thế này, để việc recognize đạt được độ chính xác cao chúng ta cần xây dựng được một bộ dataset có tính decriminitive nghĩa là có tính phân biệt giữa các thành phần. Ví dụ: các thành phần biễu diễu Person A với các thành phần biễu diễn Person B nên có sự khác biệt và các thành phần biễu diễn của Person A nên có tính tương đồng với nhau. Để làm được điều này chúng ta cần có một mô hình trích xuất ra các thuộc tính đặc trưng nhất của ảnh, chúng ta có thể tự định nghĩa một model của riêng mình hoặc sử dụng sẵn các pretrained models như faceNet, ArcFace, VggFace,... để trích xuất ra các vector đặc trưng.

* + 1. **Face Recognition**

Đây là bước cuối cùng của bài toán, để recognition chúng ta sẽ có 2 cách tiếp cận bài toán, thứ nhất là chúng ta dùng dataset trích xuất ra từ phase trước đưa vào một mạng CNN training vs labels có sẵn để cho ra một models có khả năng predict, tuy nhiên điểm hạn chế của cách này đó là nếu như bạn muốn models của bạn có khả năng predict được những person mới thì bạn phải định kỳ training lại models của mình. Vì vậy chúng ta sẽ đi theo hướng khác, ở đây cách xử lý của mình là dùng một thuật toán search dựa trên độ similarity giữa các vector đặc trưng kết hợp với công thức tính khoảng cách để cho ra được kết quả. Có thể tưởng tượng như thế này, mỗi nhóm data sẽ biễu diễn một người ở các tư thế, góc mặt khác nhau nên khi ta search với một vector input đầu vào, nó sẽ tìm tương ứng đến các nhóm data đó, nếu người đó chưa tồn tại thì sẽ có một nhóm mới được tạo ra, cơ bản là vậy.

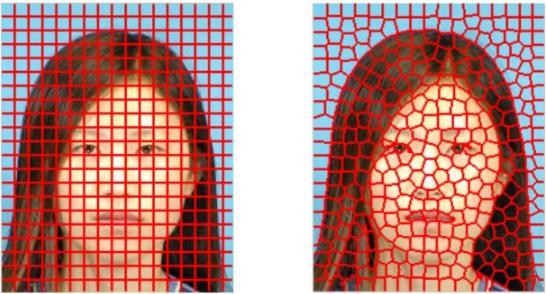
* 1. **Một số thuật toán nhận diện khuôn mặt**
     1. **Nhận Dạng Khuôn Mặt Sử Dụng Bag-of-Words**

***Tóm Tắt***

Toán khối Bag of Word nhận dạng khuôn mặt bằng cách chia khuôn mặt thành nhiều khối đặc trưng SIFT, từ đó tính toán và lượng tử hóa vector thành các codeword khác nhau. Cuối cùng, ở mỗi khối ta tính tần số phân phối của mỗi codeword, sau đó nối dài các tần số từ các khối để biểu diễn khuôn mặt.

***Nhận Xét Thuật Toán***

Từ ý tưởng chia khối vuông của thuật toán này, ta có thể thay đổi thành chia theo superpixel, tức nhóm các điểm ảnh có mức thấp thành các vùng do superpixel giữ được tính tự nhiên của ảnh và giúp tính đặc trưng ảnh tiện lợi hơn, làm giảm độ phức tạp của các quy trình xử lý ảnh sau đó. [7]

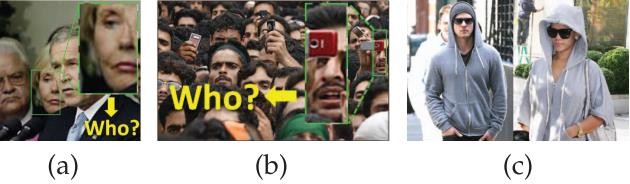
Chia ảnh theo lưới vuông (trái) và chia ảnh theo superpixel (phải).

Ta xem mỗi vùng superpixel như là một ROI, sau đó sử dụng Bag of Word để huấn luyện và phân loại khuôn mặt.

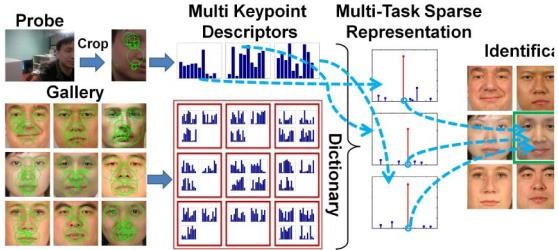
* + 1. **Nhận Dạng Một Phần Khuôn Mặt Không Cần Canh Chỉnh**

***Tóm Tắt***

Ảnh trích xuất từ camera giám sát hay camera du lịch thường xuất hiện ảnh chỉ chụp một phần mặt người. Những phương pháp nhận dạng khuôn mặt theo kiểu toàn cục (PCA và LDA) hay địa phương (Gabor, LBP) thường canh chỉnh và biểu diễn các ảnh khuôn mặt theo cùng một kích cỡ, sau đó nối dài các giá trị điểm ảnh hoặc trích xuất các vector đặc trưng theo số chiều nhất định. Tuy nhiên, với ảnh một phần khuôn mặt thì cách làm này không khả thi do mặt không đủ chi tiết cũng như sẽ gặp khó khăn khi canh chỉ mặt, với không có gì đảm bảo rằng những điểm mốc phổ biến của khuôn mặt sẽ xuất hiện ở ảnh một phần mặt. Thuật toán nhận dạng một phần mặt người mà không cần dùng tọa độ 2 mắt (hay bất kỳ điểm nào khác) để canh chỉnh mặt.

Vì dụ về ảnh một phần khuôn mặt. (a) Ảnh một phần khuôn mặt trong bộ dữ liệu LFW. (b) Ảnh một phần khuôn mặt trong đám đông. (c) Ảnh khuôn mặt bị che bởi mắt kinh, áo khoác.

Thuật toán biểu diễn khuôn mặt không cần canh chỉnh dựa trên phép Mô Tả Đa Điểm Chính (Multi Keypoint Descriptor - MKD), trong đó kích thước mô tả khuôn mặt được xác định bằng thành phần có trong ảnh. Làm theo cách này, ta có thể dùng một tập lớn các mô tả để biểu diễn bất kỳ ảnh khuôn mặt kiểm tra nào, dù là một phần hay toàn phần. Thuật toán có một phép mô tả điểm chính mới gọi là Mẫu Tam Phân Gabor (Gabor Ternary Patter - GTP) nhằm giúp nhận dạng khuôn mặt dễ dàng hơn. Thuật toán này sử dụng hiệu quả với ảnh khuôn mặt bị vật thể khá che khuất, ảnh không chính diện, ảnh có đeo phụ kiện, ảnh bị giới hạn góc nhìn, ảnh phơi sáng.

[Hình dưới](#_bookmark25) mô tả sơ đồ thuật toán sử dụng biểu diễn MKD cho thư viện từ điển cũng như ảnh kiểm tra. Sau đó, học phép Biểu Diễn Đa Nhiệm Thưa (Multi-task sparse Representation) với mỗi ảnh kiểm tra, cuối cùng, sử dụng Phân Loại dựa trên Biểu Diễn Thưa (Sparse Representation based Classification - SRC) [8] để nhận dạng ảnh. Thuật toán này gọi là MKD-SRC.

Mô tả ý tưởng cho thuật toán nhận dạng một phần khuôn mặt.

***Nhận Xét Thuật Toán***

Thuật toán MKD-SRC đã đề ra giải pháp nhận dạng khuôn mặt mà không cần canh chỉnh mặt và có thể trích xuất ra nhiều điểm chính hơn sử dụng SURF. Do đó từ các ô superpixel trong [Hình 9](#_bookmark21), ta có thể dùng MKD để lấy đặc trưng mà sử dụng Bag of Word, khi đó sẽ không cần xây dựng đến từ điển có kích thước lớn như trong MKD-SRC.

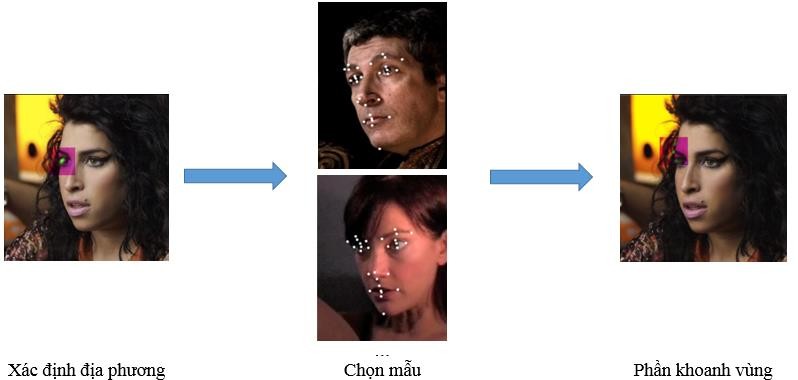
* + 1. **Khoanh Vùng Một Phần Khuôn Mặt Sử Dụng Các Mẫu Đồng Dạng**

***Tóm Tắt***

Trong nhận dạng khuôn mặt, đầu tiên cần phải xác định khuôn mặt trong bức ảnh. Các thuật toán xác định khuôn mặt thường trả về hộp chữ nhật bao quanh khuôn mặt, từ đó xác định và khoanh vùng các phần khuôn mặt như góc lông mày, góc mắt, đỉnh mũi, góc miệng, cằm, … là những điểm đặc trưng chính của khuôn mặt. Tuy nhiên, có những điểm chính không nằm ở vị trí có độ dốc cao trong ảnh (ví dụ như đỉnh mũi), và việc xác định những điểm này đòi hỏi nhiều ảnh dữ liệu. Thuật toán khoanh vùng các điểm chính đã được định sẵn trước dưới nhiều điều kiện ảnh khác nhau. Sử dụng thuật toán cho ảnh khuôn mặt chân dung, chiếu sáng, cảm xúc, kiểu tóc, tuổi của đối tượng, mặt bị che một phần.

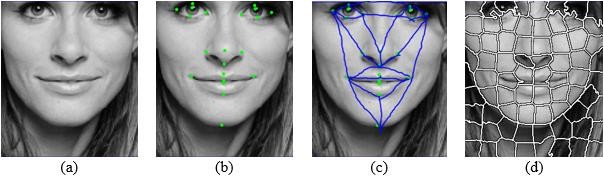
Ảnh kết quả sau khi khoanh vùng.

Việc khoanh vùng phần mặt được xem là phép suy diễn Bayes, bằng cách kết hợp output của xác định địa phương và tập phi tham số các mô hình toàn cục để khoanh vùng từng phần dựa trên hơn một ngàn mẫu ảnh đã được khoanh vùng thủ công. Giả sử rằng mô hình toàn cục tạo ra khoanh vùng từng phần là các biến ẩn, nên sử dụng hàm mục tiêu Bayes, hàm này được tối ưu hóa bằng các mẫu đồng dạng cho các biến ẩn này. Nhóm đề ra công thức xác định các phần địa phương bằng cách xác định các điểm chính có tỉ lệ mịn hơn hoặc là các đặc trưng nhỏ. Nhiều công thức xác định điểm chính sử dụng phân loại đã được huấn luyện để xác định điểm chính cụ thể (ví dụ như góc trái của mắt trái), với hàm nhân bán kính cơ sở sử dụng SVM.

Tóm tắt thuật toán. Ảnh trái: Sử dụng xác định địa phương để xác định các điểm chính (ví dụ: cạnh mắt). Ảnh giữa: Chọn mẫu từ tập huấn luyện. Ảnh phải: Kết quả khoanh vùng điểm chính.

***Nhận Xét Thuật Toán***

Thuật toán này có khả năng tìm ra các điểm chính trên khuôn mặt dưới nhiều điều kiện ảnh khác nhau, do đó ta có thể sử dụng thuật toán này hỗ trợ cho việc nhận dạng khuôn mặt. Từ ảnh đầu vào (a), đầu tiên ta xác định điểm chính trên khuôn mặt (b), từ đó tách khuôn mặt ra thành các phần mặt (c). Ngoài ra, từ khuôn mặt ban đầu, sử dụng superpixel (d), sau đó với mỗi phần mặt đã tách, chọn các superpixel nằm trong phần mặt này. Sau đó có thể áp dụng thuật toán nhận dạng khuôn mặt [2], [6].

(a): Ảnh vào. (b) Xác định các điểm chính trên khuôn mặt. (c). Từ các điểm chính, chia thành các phần của khuôn mặt (d) Từ ảnh (a), lấy superpixel, với mỗi phần đã chia từ (c), chọn superpixels nằm trong phần đó.

* + 1. **Nhận Dạng Khuôn Mặt Sử Dụng Thuật Toán DeepFace**

***Tóm Tắt***

Nhóm tác giả từ Trung tâm Nghiên cứu Facebook và trường Đại học Tel Aviv, Israel đề xuất một thuật toán có tên là DeepFace, sử dụng nguồn ảnh do người dùng đăng tải lên Facebook làm bộ dữ liệu. Với nhận dạng khuôn mặt, người ta thường trải qua 4 bước: Xác định khuôn mặt  Canh chỉnh khuôn mặt  Biểu diễn khuôn mặt  Phân loại khuôn mặt, nhóm tác giả biểu diễn khuôn mặt theo mô hình 3D nhằm áp dụng biến đổi affine từng phần, từ đó biểu diễn khuôn mặt từ 9 lớp Mạng Neuron Sâu (Deep Neural Network - DNN), mạng này có hơn 120 ngàn tham số sử dụng một số lớp liên thông mà không chia sẻ trọng số.

Nhóm tác giả phát triền một cấu trúc DNN hiệu quả và các tận dụng ảnh trên mạng xã hội để biểu diễn khuôn mặt sao cho có thể tổng quát hóa cho các tập dữ liệu khác. Ngoài ra, nhóm tác giả trình bày cách canh chỉnh mặt dựa trên mô hình 3D của khuôn mặt.

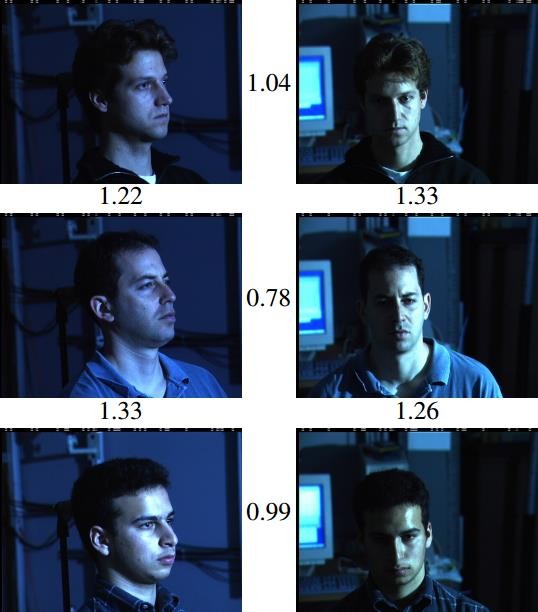
***Nhận Xét Thuật Toán***

DeepFace đã đưa ra một cấu trúc mạng neuron sử dụng mô hình 3D của khuôn mặt, từ đó giúp canh chỉnh khuôn mặt về chính diện. Do đó, trong bài toán nhận dạng một phần khuôn mặt, từ bộ dữ liệu, ta có thể xây dựng cấu trúc 3D của khuôn mặt, sau đó với ảnh kiểm tra với góc mặt tùy ý, ta có thể áp lên mô hình 3D này để ước lượng mặt chính diện của ảnh kiểm tra.

* + 1. **Nhận Dạng Khuôn Mặt Sử Dụng Thuật Toán FaceNet**

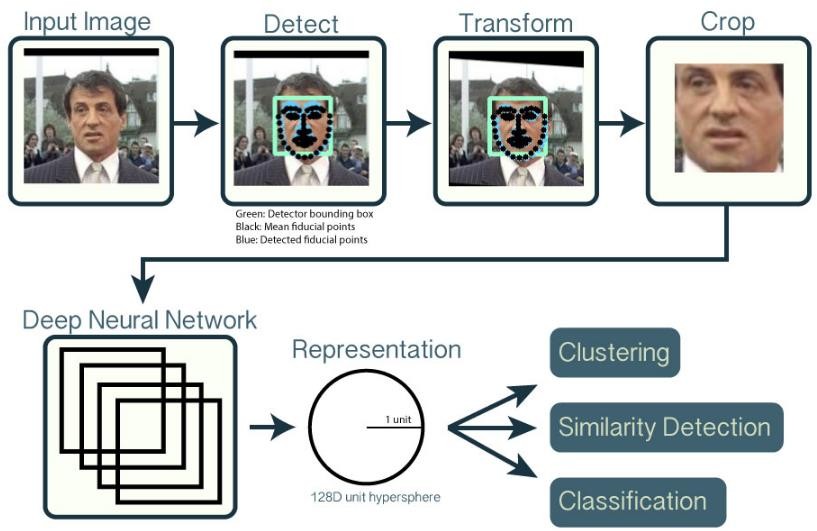
***Tóm Tắt***

Nhóm tác giả từ Google đề xuất một thuật toán có tên là FaceNet sẽ học cách ánh xạ từ ảnh khuôn mặt vào không gian Euclide compact với khoảng cách đo được tương ứng với độ tương đồng của khuôn mặt. Thuật toán này có thể tạo ra vector đặc trưng và nhúng vào bài toán nhận dạng khuôn mặt, kiểm tra khuôn mặt và phân cụm khuôn mặt. Nhóm tác giả sử dụng Mạng Tích Chập Sâu (Deep Convolution Network - DNN) được huấn luyện để tự tối ưu hóa bài toán. Mạng được huấn luyện sao cho khoảng cách L2 bình phương trong không gian nhúng tương ứng với mức độ tương đồng của khuôn mặt: Mặt cùng người sẽ có khoảng cách nhỏ, mặt khác người sẽ có khoảng cách lớn.



Hình minh họa output khoảng cách khi sử dụng FaceNet giữa các cặp khuôn mặt. Nếu lấy ngưỡng là 1.1, ta thấy rằng 2 mặt có khoảng cách nhỏ hơn ngưỡng đều thuộc về một người (ví dụ như 2 ảnh ở dòng đầu tiên) và ngược lại.

Sau khi thực hiện phép nhúng, thu được vector đặc trưng thì ta có thể thực hiện được 3 bài toán: Kiểm tra khuôn mặt, ta chỉ cần phân ngưỡng khoảng cách giữa 2 vector đặc trưng của 2 khuôn mặt. Nhận dạng khuôn mặt là bài toán phân loại k-NN. Phân cụm khuôn mặt sử dụng k- mean.



Tóm tắt quy trình nhận dạng khuôn mặt sử dụng FaceNet,từ ảnh vào (input image), sau đó xác định khuôn mặt, những điểm chính trên mặt (Detect), canh chỉnh lại mặt (Transform), sau đó cắt khuôn mặt ra khỏi ảnh (Crop) và đưa vào Mạng Neuron Sâu (Deep Neural Network), thu được vector đặc trưng 128 chiều dùng để biểu diễn khuôn mặt (Representation). Từ vector đặc trưng này có thể dùng để phân cụm khuôn mặt (Clustering), xác định tính tương đồng (Similarity Detection) và phân loại (Classification).

Nhiều thuật toán nhận dạng khuôn mặt sử dụng DNN trước đây sử dụng lớp phân loại đã qua huấn luyện trên toàn bộ ảnh đã biết nhãn, sau đó lấy lớp thắt cổ chai trung bình để biểu diễn tổng quát cho tập huấn luyện. Tuy nhiên, mặt tiêu cực là lớp thắt cổ chai đôi khi không rõ 26as và không tiện lợi do lớp thắt cổ chai này thường rất lớn (khoảng 1000 chiều). Để khắc phục điều này, FaceNet huấn luyện output thành nhúng compact 128 chiều sử dụng hàm bộ ba sai số dựa trên LMNN, mẫu bộ ba này gồm 2 ảnh cùng loại và 1 ảnh khác loại và hàm sai số có nhiệm vụ tách ảnh đúng ra khỏi ảnh sai dựa vào biên khoảng cách. Nhóm tác giả sử dụng 2 kiến trúc Mạng Tích Chập Sâu, một mạng dựa theo mô hình của Zeiler và Fergus, mạng còn lại sử dụng mô hình Inception từ GoogLeNet.

***Nhận Xét Thuật Toán***

Thuật toán FaceNet sử dụng bộ ba sai số và CNN để huấn luyện, có thể sử dụng ý tưởng này vào đề tài. Tuy nhiên, vấn đề gặp phải là ta không có đủ thiết bị để huấn luyện triệu ảnh như FaceNet. Do đó, thay vì huấn luyện trên toàn bộ khuôn mặt, ta có thể huấn luyện từng phần trên khuôn mặt dựa trên ý tưởng trình bày ở mục [2.3.5](#_bookmark62).

* 1. **Thuật toán MTCNN và Facenet của dự án**
     1. **Lý do chọn Facenet**

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.

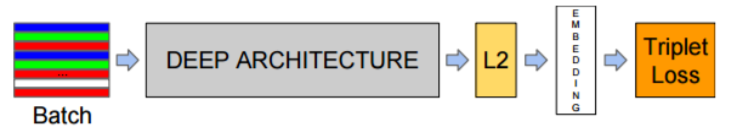
* Tuy nhiên hạn chế của các thuật toán này đó là số lượng chiều embedding tương đối lớn (thường >= 1000) và ảnh hưởng tới tốc độ của thuật toán. Thường chúng ta phải áp dụng thêm thuật toán PCA để giảm chiều dữ liệu để giảm tốc độ tính toán.
* Hàm loss function chỉ đo lường khoảng cách giữa 2 bức ảnh. 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.

Facenet được Google giới thiệu vào năm 2015, và model này thì cho ảnh vào (đúng size) thì trả ra 1 vector 128 features cho 1 khuôn mặt. Sau đó dùng SVM để phân nhóm các vector đó vào các nhóm để biết vector đó là mặt của ai.

* + 1. **Chi Tiết Thuật Toán Facenet**



Cấu trúc mô hình, mạng bao gồm khối lớp vào (batch), đi qua cấu trúc Mạng Neuron Tích Chập Sâu (Convolution Neural Network – CNN), sau đó chuẩn hóa theo 𝐿2 và đưa vào hệ thống nhúng. Trong quá trình huấn luyện có sử dụng bộ ba sai số.

FaceNet sử dụng DNN, giả sử cấu trúc mô hình là một khối lớp (xem Hình 30), sau khi sử dụng cấu trúc CNN, vấn đề quan trọng nằm ở kết quả sau khi huấn luyện. Do đó, nhóm tác giả sử dụng đến bộ ba sai số có thể giúp kiểm tra, nhận dạng và phân cụm khuôn mặt. Giả sử ta có ảnh 𝑥, đưa qua hàm nhúng 𝑓(𝑥) vào không gian đặc trưng ℝ𝑑 sao cho khoảng cách bình phương của tất cả khuôn mặt cùng loại phải nhỏ hơn khoảng cách bình phương với mặt khác loại.

Bộ Ba Sai Số

Ta biểu diễn phép nhúng là hàm 𝑓(𝑥) ∈ ℝ𝑑 có chức năng nhúng ảnh 𝑥 vào không gian Euclide 𝑑 chiều. Hơn nữa, ta xét hàm nhúng này xác định trong siêu cầu 𝑑 chiều, tức ‖𝑓(𝑥)‖2 = 1. Giả sử 𝑥𝑖 𝑎 là ảnh kiểm tra người 𝑖, 𝑥𝑖 𝑝 là ảnh của người 𝑖 trong bộ dữ liệu và 𝑥𝑖 𝑛 là ảnh không phải của người 𝑖 (xem Hình 31), ta muốn rằng khoảng cách từ 𝑥𝑖 𝑎 đến 𝑥𝑖 𝑝 phải ngắn hơn khoảng cách từ 𝑥𝑖 𝑎 đến 𝑥𝑖 𝑛 , tức 𝑑(𝑥𝑖 𝑎 , 𝑥𝑖 𝑝 ) < 𝑑(𝑥𝑖 𝑎 , 𝑥𝑖 𝑛 ).

Do đó, ta muốn ‖𝑓(𝑥𝑖 𝑎 ) − 𝑓(𝑥𝑖 𝑝 )‖ 2 2 + 𝛼 < ‖𝑓(𝑥𝑖 𝑎 ) − 𝑓(𝑥𝑖 𝑛 )‖2 2 , (30) ∀(𝑓(𝑥𝑖 𝑎 ), 𝑓(𝑥𝑖 𝑝 ), 𝑓(𝑥𝑖 𝑛 )) ∈ 𝑇 (31) với 𝛼 là giá trị biên sao cho đảm bảo bất đẳng thức (30) xảy ra, 𝑇 là tập các mẫu bộ 27as a thể xảy ra trong tập huấn luyện. Ta cực tiểu hóa hàm sai số như sau 𝐿 = ∑[‖𝑓(𝑥𝑖 𝑎 ) − 𝑓(𝑥𝑖 𝑝 )‖ 2 2 − ‖𝑓(𝑥𝑖 𝑎 ) − 𝑓(𝑥𝑖 𝑛 )‖2 2 + 𝛼] + 𝑁 𝑖 (32) Có rất nhiều bộ ba thỏa (30), những bộ ba này không đóng góp nhiều vào quá trình huấn luyện và khiến cho tốc độ hội tụ chậm. Do đó, ta cần chọn bộ ba thích hợp cho quá trình huấn luyện, giữ vai trò quan trọng trong mô hình.

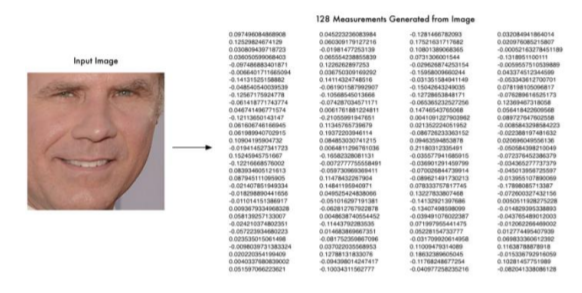
Chọn Bộ Ba

Để đảm bảo quá trình huấn luyện hội tụ nhanh, ta sẽ chọn bộ ba không thỏa bất đẳng thức (30), tức cho ảnh 𝑥𝑖 𝑎 của người 𝑖, ta chọn 𝑥𝑖 𝑝 sao cho arg max 𝑥𝑖 𝑝 ‖𝑓(𝑥𝑖 𝑎 ) − 𝑓(𝑥𝑖 𝑝 )‖ 2 2 (33) và chọn 𝑥𝑖 𝑛 sao cho arg min 𝑥𝑖 𝑛 ‖𝑓(𝑥𝑖 𝑎 ) − 𝑓(𝑥𝑖 𝑛 )‖2 2 (34) 28 Điều này có nghĩa trong tập ảnh cùng đối tượng với 𝑥𝑖 𝑎 , ta chọn ảnh 𝑥𝑖 𝑝 (hard positive) sao cho khoảng cách giữa chúng là lớn nhất và trong tập ảnh khác đối tượng với 𝑥𝑖 𝑎 , chọn ảnh 𝑥𝑖 𝑛 (hard negative) sao cho khoảng cách giữa chúng là nhỏ nhất, khi đó có khả năng xảy ra trường hợp ‖𝑓(𝑥𝑖 𝑎 ) − 𝑓(𝑥𝑖 𝑝 )‖ 2 2 > ‖𝑓(𝑥𝑖 𝑎 ) − 𝑓(𝑥𝑖 𝑛 )‖2 2 (35) Ta sẽ huấn luyện sao cho bất đẳng thức (35) về lại kiểu (30). Hình 32 cho thấy quy trình huấn luyện sẽ rút ngắn khoảng cách giữa 𝑥𝑖 𝑎 và 𝑥𝑖 𝑝 xuống thấp nhất và kéo dài khoảng cách giữa 𝑥𝑖 𝑎 và 𝑥𝑖 𝑛 ra xa nhất

Không thể tính arg min và arg max trên toàn bộ tập huấn luyện vì có thể đưa ra kết quả huấn luyện kém và quá trình huấn luyện có thể lấy ảnh có chất lượng kém làm hard positive và hard negative. Để khắc phục tình trạng này, nhóm tác giả sử dụng khối mini lớn với vài ngàn mẫu, dùng khối này vào quá trình huấn luyện, tạo bộ ba sau mỗi 𝑛 bước huấn luyện, chọn điểm checkpoint mới nhất ở trong mạng và tính arg min và arg max trong tập dữ liệu con. Để biểu diễn khoảng cách giữa 𝑥𝑖 𝑎 và 𝑥𝑖 𝑝 có nghĩa, ta cần đảm bảo tối thiểu các mẫu từ tất cả đối tượng phải có trong khối mini. Khi thực nghiệm, nhóm tác giả lấu mẫu dữ liệu huấn luyện sao cho mỗi đối tượng có 40 ảnh ở mỗi khối mini. Hơn nữa, chọn ngẫu nhiên ảnh 𝑥𝑖 𝑛 và thêm vào mỗi khối. Thay vì chọn ảnh cùng loại có khoảng cách xa nhất, nhóm tác giả sử dụng tất cả cặp (𝑥 𝑎 , 𝑥 𝑝) trong khối mini, đồng thời tìm kiếm ảnh hard negative. Thực nghiệm cho thấy cặp (𝑥 𝑎 , 𝑥 𝑝) có tính ổn định và hội tụ nhanh. Chọn mẫu khác loại có khoảng cách gần nhất có thể đưa đến lỗi cực tiểu địa phương khi huấn luyện, dễ dẫn đến mô hình bị sập (tức 𝑓(𝑥) = 0). Để tránh trường hợp này, ta chọn 𝑥𝑖 𝑛 sao cho ‖𝑓(𝑥𝑖 𝑎 ) − 𝑓(𝑥𝑖 𝑝 )‖ 2 2 < ‖𝑓(𝑥𝑖 𝑎 ) − 𝑓(𝑥𝑖 𝑛 )‖2 2 (36) Ta gọi các mẫu 𝑥𝑖 𝑛 thỏa bất đẳng thức (36) là semi-hard do khoảng cách từ mẫu này đến 𝑥𝑖 𝑎 xa hơn khoảng cách đến 𝑥𝑖 𝑝 , nhưng ta vẫn gặp khó khăn do bình phương khoảng cách sát với khoảng cách 𝑥𝑖 𝑎 đến 𝑥𝑖 𝑝 . Các mẫu 𝑥𝑖 𝑛 này nằm trong biên 𝛼. Chọn đúng bộ ba sẽ giúp quá trình huấn luyện hội tụ nhanh. Mặt khác, nhóm tác giả sử dụng khối mini nhỏ do khối này giúp cải thiện khả năng hội tụ khi sử dụng kỹ thuật Trượt Dốc Ngẫu Nhiên (Stochastic Gradient Descent – SGD) [28] bằng cách lấy phần biểu thức trong dấu Σ ở (32) ‖𝑓(𝑥𝑖 𝑎 ) − 𝑓(𝑥𝑖 𝑝 )‖ 2 2 − ‖𝑓(𝑥𝑖 𝑎 ) − 𝑓(𝑥𝑖 𝑛 )‖2 2 + 𝛼 (37) sau đó lấy đạo hàm hàm 𝐿 theo từng biến 𝑥𝑖 𝑎 , 𝑥𝑖 𝑝 , 𝑥𝑖 𝑛 29 𝜕𝐿 𝜕𝑥𝑖 𝑎 = ∑{ 2 (𝑓(𝑥𝑖 𝑛 ) − 𝑓(𝑥𝑖 𝑝 )) , nếu (37) ≥ 0 0, ngược lại 𝑁 𝑖=1 (38) 𝜕𝐿 𝜕𝑥𝑖 𝑝 = ∑{ −2 (𝑓(𝑥𝑖 𝑎 ) − 𝑓(𝑥𝑖 𝑝 )) , nếu (37) ≥ 0 0, ngược lại 𝑁 𝑖=1 (39) 𝜕𝐿 𝜕𝑥𝑖 𝑛 = ∑{ −2(𝑓(𝑥𝑖 𝑎 ) − 𝑓(𝑥𝑖 𝑛 )), nếu (37) ≥ 0 0, ngược lại 𝑁 𝑖=1 (40) sau đó, ta cập nhật tham số hàm sai số 𝑓(𝑥𝑖 𝑎 ) = 𝑓(𝑥𝑖 𝑎 ) − 𝛿 𝜕𝐿 𝜕𝑥𝑖 𝑎 (41) 𝑓(𝑥𝑖 𝑝 ) = 𝑓(𝑥𝑖 𝑝 ) − 𝛽 𝜕𝐿 𝜕𝑥𝑖 𝑝 (42) 𝑓(𝑥𝑖 𝑛 ) = 𝑓(𝑥𝑖 𝑛 ) − 𝛾 𝜕𝐿 𝜕𝑥𝑖 𝑛 (43) với 𝛿, 𝛽, 𝛾 là tốc độ học. Khi thực nghiệm, nhóm tác giả sử dụng khối gồm 1800 mẫu.

Mạng Tích Chập Sâu

Khi thực nghiệm, nhóm tác giả huấn luyện sử dụng CNN sử dụng SGD với kỹ thuật truyền ngược chuẩn [29] [30] và AdaGrad [31]. Khi thực nghiệm, nhóm chọn tốc độ học 𝛿 = 𝛽 = 𝛾 = 0.05, huấn luyện qua một cụm CPU trong 1000 giờ đến 2000 giờ, hàm chi phí giảm dần (tức độ chính xác tăng dần) sau 500 giờ huấn luyện. Sau huấn luyện. mỗi ảnh trả về vector đặc trưng 128 chiều (xem Hình 33), tính chất với 2 ảnh cùng người thì khoảng cách hai vector gần hơn khoảng cách với ảnh của người khác.

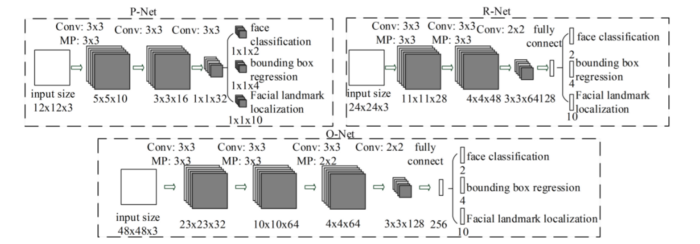
Ảnh vào (trái) sau khi huấn luyện, thu được vector 128 chiều (phải)

Nhóm tác giả sử dụng 2 kiến trúc để học, một kiến trúc của Zeiler và Fergus [26. Nhóm tác giả [23] thêm lớp tích chập 1 × 1 × 𝑑, thu được mô hình sâu 22 lớp với 140 triệu tham số và cần 1.6 tỷ toán tử/giây/ảnh.

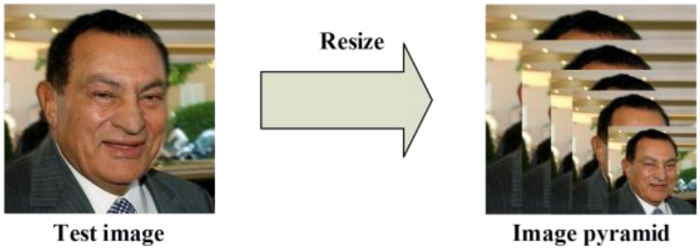
MTCNN

MTCNN là viết tắt của Multi-task Cascaded Convolutional Networks. Nó là bao gồm 3 mạng CNN xếp chồng và đồng thời hoạt động khi detect khuôn mặt. Mỗi mạng có cấu trúc khác nhau và đảm nhiệm vai trò khác nhau trong task. Đầu ra của MTCNN là vị trí khuôn mặt và các điểm trên mặt như: mắt, mũi, miệng…

MTCNN model bao gồm 3 mạng riêng biệt: P-Net, R-Net và O-Net.



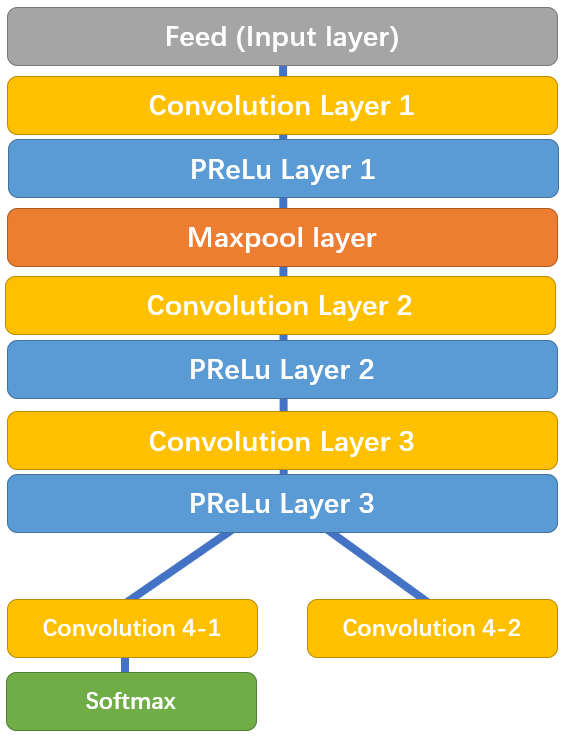
Với mỗi input image, mạng sẽ tạo ra image pyramid.



P-Net[Permalink](https://minhdq99hp.github.io/programming/mtcnn/#p-net)

Trong mạng P-net, với từng scaled image đó, 12x12 kernel sẽ trượt trên ảnh để tìm mặt.

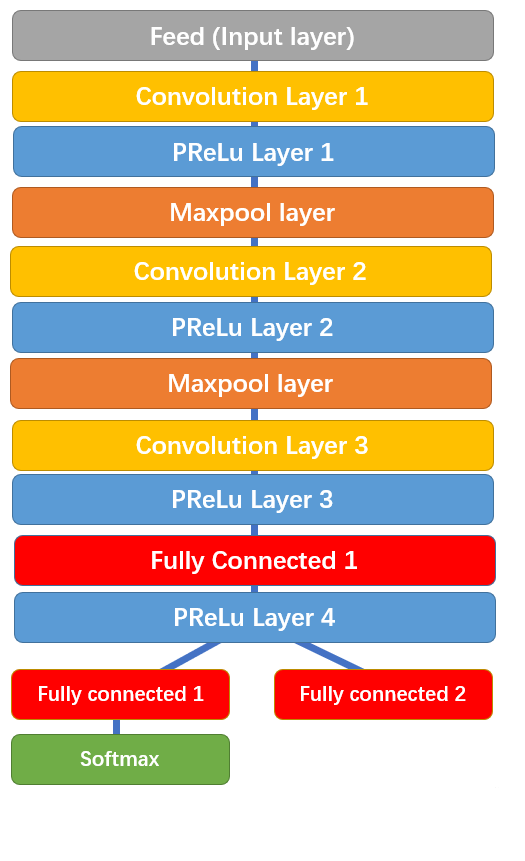




Sau PReLu Layer 3 thì chia làm 2 nhánh:

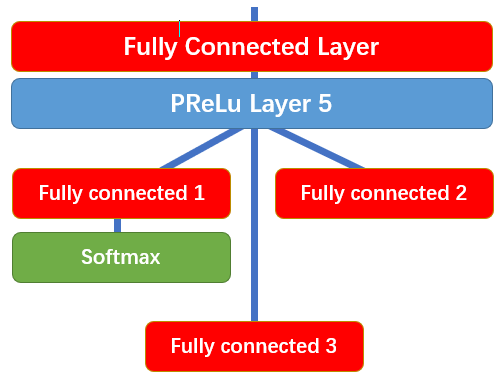
* Convolution 4-1: probability của mặt nằm trong mỗi bbox.
* Convolution 4-2: tọa độ của các bbox.

R-Net[Permalink](https://minhdq99hp.github.io/programming/mtcnn/#r-net)

Tương tự, cũng chia làm 2 nhánh như P-Net.

O-Net[Permalink](https://minhdq99hp.github.io/programming/mtcnn/#o-net)

O-Net lấy đầu ra của R làm đầu vào và dùng để xác định tọa độ các facial landmarks.



O-Net chia làm 3 nhánh xác định:

* Tọa độ của bbox
* Tọa độ của facial landmarks
* Xác suất mặt nằm trong bbox đó.

Các bảng sau đây cho thấy điểm chuẩn của việc triển khai mtcnn này chạy trên [CPU Intel i7-3612QM @ 2.10GHz](https://www.cpubenchmark.net/cpu.php?cpu=Intel+Core+i7-3612QM+%40+2.10GHz) , với Tensorflow 1.4.1 **dựa trên CPU** .

* *Hình ảnh có một mặt chính diện:*

| **Kích cỡ hình** | **Tổng số pixel** | **Tiến trình thời gian** | **FPS** |
| --- | --- | --- | --- |
| 460x259 | 119.140 | 0,118 giây | 8.5 |
| 561x561 | 314.721 | 0,227 giây | 4,5 |
| 667x1000 | 667.000 | 0,456 giây | 2,2 |
| 1920x1200 | 2.304.000 | 1,093 giây | 0,9 |
| 4799x3599 | 17.271.601 | 8.798 giây | 0,1 |

* *Hình ảnh có 10 mặt chính diện:*

| **Kích cỡ hình** | **Tổng số pixel** | **Tiến trình thời gian** | **FPS** |
| --- | --- | --- | --- |
| 474x224 | 106.176 | 0,185 giây | 5,4 |
| 736x348 | 256.128 | 0,290 giây | 3,4 |
| 2100x994 | 2.087.400 | 1,286 giây | 0,7 |

* Theo mặc định, MTCNN bao gồm một mô hình trọng số phát hiện khuôn mặt.
* Mô hình này được điều chỉnh từ việc triển khai MTCNN của Facenet, được hợp nhất trong một tệp duy nhất nằm bên trong thư mục 'dữ liệu' liên quan đến đường dẫn của mô-đun. Nó có thể được ghi đè bằng cách đưa nó vào hàm khởi tạo MTCNN () trong quá trình khởi tạo.
* Mô hình phải dựa trên numpy chứa 3 khóa chính “pnet”, “rnet” và “onet”, mỗi khóa có trọng số của mỗi lớp trong mạng.

Thực nghiệm

Nhóm tác giả đánh giá trên bộ dữ liệu LFW và Youtube Faces và thực nghiệm 3 vấn đề: Nhận dạng khuôn mặt, kiểm tra khuôn mặt và phân cụm khuôn mặt. Do mục tiêu báo cáo nói về nhận dạng khuôn mặt nên chỉ trình bày kết quả thực nghiệm nhận dạng khuôn mặt khi sử dụng FaceNet. Khi huấn luyện, nhóm tác giả sử dụng 100 triệu đến 200 triệu ảnh khuôn mặt từ 8 triệu đối tượng, sau khi cắt phần khuôn mặt trong ảnh để tạo thành ảnh khuôn mặt, nhóm thay đổi kích thước ảnh khuôn mặt từ 96 × 96 điểm ảnh đến 224 × 224 điểm ảnh. Với bài toán nhận dạng khuôn mặt, sau khi huấn luyện thu được vector đặc trưng 128 chiều thì ta sử dụng phân loại k-NN. Nhóm đánh giá trên bộ dữ liệu LFW với 13233 ảnh khuôn mặt từ 5749 người và thu được độ chính xác 98.87% ± 0.15 khi không canh chỉnh mặt và 99.63% ± 0.09 khi có canh chỉnh mặt. Hình 37 là một số cặp ảnh nhận dạng lỗi trong bộ LFW.



Một số cặp ảnh nhận dạng sai trong bộ dữ liệu LFW.

Trong bộ dữ liệu Youtube Face bao gồm 3425 video với 1595 người, trong 100 frame đầu tiên ở mỗi video, nhóm tác giả xác định khuôn mặt, tính trung bình tương đương cho tất cả cặp 32 khuôn mặt, khi đó độ chính xác là 95.12% ± 0.39. Nếu sử dụng 1000 frame đầu tiên thì độ chính xác là 95.18%.

* + 1. **Ưu điểm và nhược điểm của thuật toán**
* **Ưu Điểm**: Tính đến thời điểm FaceNet ra đời, thuật toán này đã lập nên kỷ lục mới trong nhận dạng khuôn mặt dưới nhiều điều kiện ảnh khác nhau.
* **Nhược Điểm:** FaceNet huấn luyện với một số lượng lớn hình ảnh (hơn 200 triệu ảnh của 8 triệu đối tượng), lớn gấp 3 lần so với các bộ dữ liệu hiện có. Để xây dựng bộ dữ liệu lớn như vậy rất khó thực hiện trong các phòng thiết bị, học thuật do đòi hỏi kiến trúc máy lớn.
* **Nhận Xét Thuật Toán:** Thuật toán FaceNet sử dụng bộ ba sai số và CNN để huấn luyện, có thể sử dụng ý tưởng này vào đề tài. Tuy nhiên, vấn đề gặp phải là ta không có đủ thiết bị để huấn luyện triệu ảnh như FaceNet. Do đó, thay vì huấn luyện trên toàn bộ khuôn mặt, ta có thể huấn luyện từng phần trên khuôn mặt.
  + 1. **So sánh thuật toán Facenet và LBPH trong nhận diện khuôn mặt**

|  |  |  |
| --- | --- | --- |
|  | Facenet | LBPH |
| Độ chính xác | Cao, có thể nhận diện ở nhiều góc cạnh của khuôn mặt. | Thấp, phụ thuộc vào nhiều vị trí khuôn mặt và ánh sáng. |
| Tốc độ xử lý | Phát hiện và nhận diện nhanh, có thể nói real time | Phát hiện và nhận diện chậm. Bị delay. |
| Triển khai thuật toán | Khó, cần sử dụng nhiều model, | Dễ triển khai. |
| Áp dụng thực tế | Nhiều công ty đã áp dụng thuật toán Facenet để tạo ra các sản phẩm như điểm danh, chấm công,.. | Không áp dụng nhiều vào thực tế. |