ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA TOÁN – TIN HỌC

Học phần: Seminar Phương pháp Toán trong Tin học
Mã học phần: TTH510

NHẬN DẠNG KHUÔN MẶT NGƯỜI DỰA TRÊN MỘT PHẦN THÔNG TIN KHUÔN MẶT

GIẢNG VIÊN HƯỚNG DẪN: PGS. TS. Phạm Thế Bảo

SINH VIÊN THỰC HIỆN: Võ Hoàng Trọng – 1311372

TP. HỒ CHÍ MINH, NGÀY 16 THÁNG 01 NĂM 2017

MỤC LỤC

DANH MỤC HÌNH

LỜI MỞ ĐẦU

Bài báo cáo này trình bày về một số thuật toán cũng như hướng phát triển cho đề tài "Nhận dạng khuôn mặt người dựa trên một phần thông tin khuôn mặt". Nội dung bài báo cáo này gồm 4 phần:

1. Giới thiệu đề tài: Trình bày sơ nét về đề tài nhận dạng khuôn mặt, những ứng dụng cũng như yêu cầu đề tài "Nhận dạng khuôn mặt người dựa trên một phần thông tin khuôn mặt".
2. Các công trình liên quan: Trình bày 5 thuật toán liên quan đến đề tài gồm nhận dạng khuôn mặt sử dụng Bag of Word, nhận dạng khuôn mặt không cần canh chỉnh, khoanh vùng một phần khuôn mặt sử dụng các mẫu đồng dạng, nhận dạng khuôn mặt sử dụng thuật toán FaceNet và nhận dạng khuôn mặt sử dụng thuật toán DeepFace. Mỗi thuật toán bao gồm 5 phần:
    a. Tóm tắt: Khái quát về thuật toán.
    b. Chi tiết thuật toán: Phân tích chi tiết bên trong thuật toán.
    c. Kết quả thực nghiệm: Trình bày kết quả thực nghiệm thuật toán trên các bộ dữ liệu và độ chính xác đạt được.
    d. Ưu và nhược điểm của thuận toán: Phân tích ưu và nhược điểm của thuật toán dựa trên ý kiến từ các bài báo quốc tế có trích dẫn thuật toán.
    e. Nhận xét thuật toán: Nhận xét của bản thân về thuật toán và cách áp dụng vào đề tài.
3. Bộ dữ liệu sử dụng cho đề tài: Trình bày 3 bộ dữ liệu dùng cho đề tài gồm bộ dữ liệu PIE, UMIST và CVL.
4. Hướng phát triển tiếp theo: Trình bày 2 hướng phát triển tiếp theo của đề tài vào luận văn tốt nghiệp.

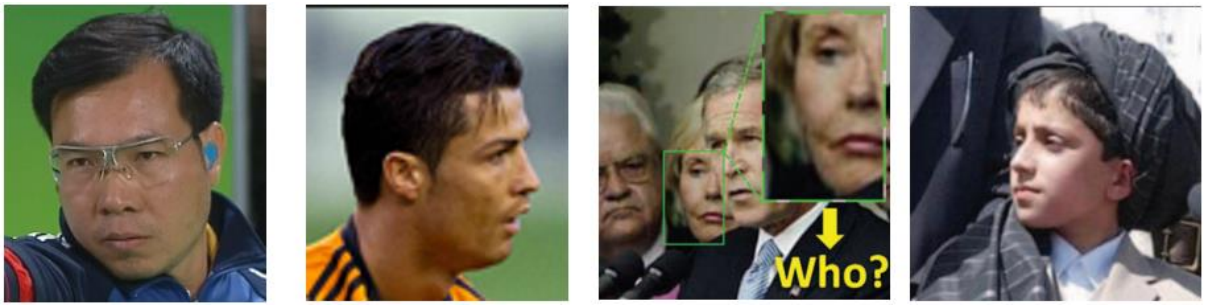
1. GIỚI THIỆU ĐỀ TÀI

1.1. Tổng Quan về Nhận Dạng Khuôn Mặt

Nhận dạng khuôn mặt là một bài toán lâu đời và được nghiên cứu rộng rãi trong khoảng hơn 30 năm trở lại đây. Bài toán nhận dạng khuôn mặt có thể áp dụng rộng rãi trong nhiều lĩnh vực khác nhau. Các ứng dụng liên quan đến nhận dạng khuôn mặt có thể kể như: Hệ thống phát hiện tội phạm, hệ thống theo dõi nhân sự trong một đơn vị, hệ thống tìm kiếm thông tin trên ảnh, video dựa trên nội dung, … Hiện nay, bài toán nhận dạng khuôn mặt gặp nhiều thách thức, ví dụ như hệ thống camera công cộng, chụp hình vui chơi thì ảnh mặt nhận được có thể bị che khuất một phần, ảnh chụp không chính diện hay chất lượng ảnh không tốt, những yếu tố này ảnh hưởng không nhỏ đến các thuật toán nhận dạng khuôn mặt. Có nhiều thuật toán khắc phục điều này, họ sử dụng một số kỹ thuật như xác định nhiều điểm chính trên khuôn mặt, lấy những chi tiết nhỏ hay sử dụng các phương pháp Học Sâu. Bài báo cáo này sẽ trình bày 5 thuật toán, trong đó 4 thuật toán nhận dạng khuôn mặt và 1 thuật toán xác định điểm chính trên khuôn mặt, những thuật toán này có thể hỗ trợ vào đề tài.

1.2. Yêu Cầu Đề Tài

Từ ảnh chụp một phần (hay một góc) của khuôn mặt, ta cần xác định xem mặt đó là của ai. Yêu cầu ảnh phải đảm bảo thấy được ít nhất 50% diện tích khuôn mặt và ít nhất một phần chi tiết ở mắt, mũi, miệng (xem Hình 1).



Hình 1 Ảnh chụp một phần khuôn mặt, ta cần xác định mặt đó là ai

2. CÁC CÔNG TRÌNH LIÊN QUAN

2.1. Nhận Dạng Khuôn Mặt Sử Dụng Bag-of-Words

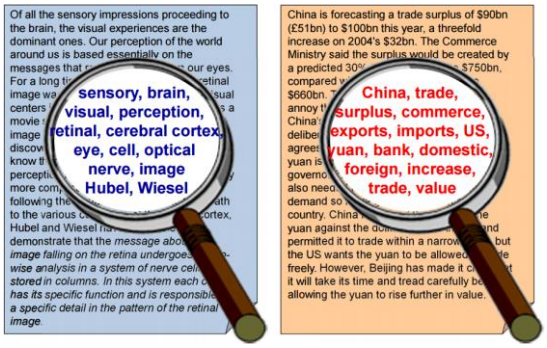
2.1.1. Tóm Tắt

Nhóm tác giả [1] đề xuất một thuật toán khối Bag of Word để 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.

2.1.2. Chi Tiết Thuật Toán

2.1.2.1. Bag of Word

Mô hình Bag of Word được sử dụng đầu tiên vào bài toán phân tích văn bản, sau đó áp dụng vào thị giác máy tính [2]. Trong bài toán phân tích văn bản, Bag of Word sẽ phân tích văn bản để thu được các "từ khóa", hay codebook, tập hợp các codebook sẽ bỏ vào một cái "túi" (bag) và ta xem cái túi này chứa đựng các từ khóa đặc trưng cho văn bản đó.



Hình 2 Với mỗi văn bản, ta thu được các từ khóa đặc trưng tương ứng, gọi là codebook, ta cho các codebook này vào một cái túi. Ví dụ như văn bản ở hình bên phải, có codebook trong kính lúp là "China", "trade", …

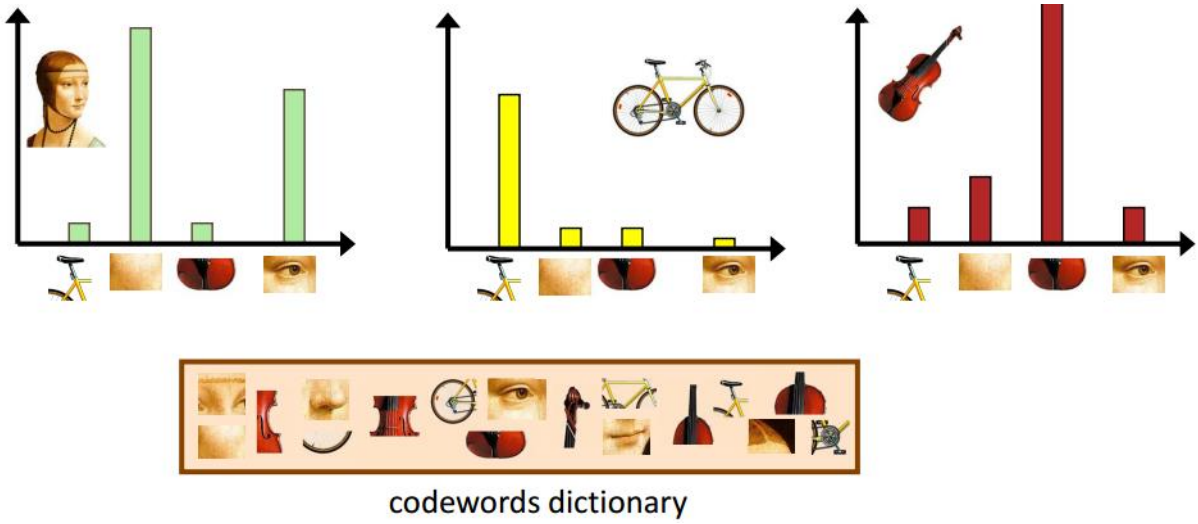
Với nhiều văn bản, tập hợp lại các túi codebook sẽ thu được từ điển codeword. Giả sử ta có một cụm từ khóa, với mỗi từ khóa, ta đối chiếu với các túi codebook có trong từ điển codeword, nếu túi của codebook nào có số lần xuất hiện nhiều nhất, ta có thể xem cụm từ trên tương ứng với văn bản của túi codebook đó, ví dụ như Hình 2, với cụm từ khóa trong kính lúp bên trái, ta có thể tìm ra văn bản tương ứng. [3]

Áp dụng ý tưởng Bag of Word vào thị giác máy tính [4]. Trong bài toán nhận dạng vật thể, ta muốn máy tính có thể tự nhận dạng được đâu là cái đồng hồ, đâu là cái TV, tủ lạnh, … Từ ý tưởng của Bag of Word, ta tìm ra các "codebook" trong mỗi vật thể, sau đó cho vào cái túi đặc trưng của vật thể, ví dụ như Hình 3 và Hình 4.

Hình 3 Với hình cô gái bên trái, ta có đặc trưng là mắt, mũi, miệng, cằm, tóc .. sau đó cho vào một cái túi đặc trưng như hình bên phải.

Hình 4 Với mỗi vật thể như khuôn mặt (trái), xe đạp (giữa), đàn violin (phải), ta thu được các đặc trưng tương ứng.

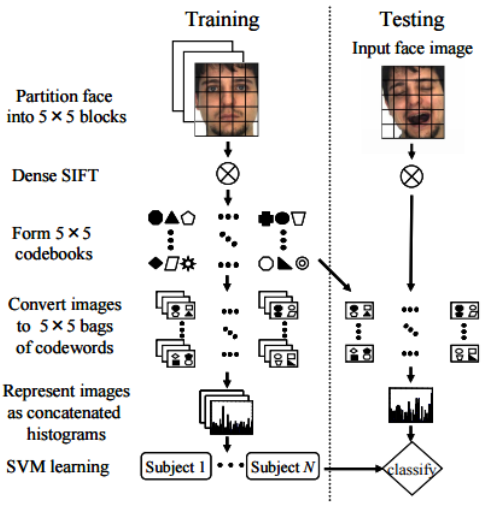
Khi nhận dạng một vật thể, máy tính xác định các đặc trưng của vật thể, ta tính tần số xuất hiện của từng đặc trưng với các codebook trong từ điển codeword, túi của codebook nào có số lần xuất hiện nhiều nhất, ta sẽ xác định được vật thể đó là gì (xem Hình 5).



Hình 5 Với mỗi vật thể trong 3 hình ở trên, ta lấy đặc trưng tương ứng, sau đó đối chiếu với từ điển codeword (hình dưới), từ đó xác định được vật thể đó là gì.

2.1.2.2. Áp Dụng Khối Bag of Word vào Nhận Dạng Khuôn Mặt

Nhóm tác giả [2] đánh giá rằng các ảnh khuôn mặt đều cùng một loại vật thể, cho nên nếu ta trính xuất đặc trưng khuôn mặt bằng cách thành tập các phần nhỏ thì điều này không đảm bảm rõ thông tin của khuôn mặt. Do đó, nhóm đề xuất thuật toán rút trích đặc trưng khuôn mặt như Hình 6.



Hình 6 Sơ đồ thuật toán Khối Bag of Word.

Ta chia ảnh thành các khối 5×5 và xem mỗi khối nhỏ là vùng quan tâm (ROI – Region of Interest). Với mỗi ROI, ta tính đặc trưng SIFT đặc trên mỗi đoạn lấy mẫu dài 2 điểm ảnh, thu được vector SIFT 128 chiều, từ đó, mỗi khối ta thu được một tập các vector SIFT. Ở bước huấn luyện, sử dụng thuật toán k-means chuyển đổi vector SIFT ở mỗi ROI thành các codeword. Ở trong một ROI, ta phân vùng các đặc trưng SIFT ở mỗi đoạn thành K cụm, khi đó ta định nghĩa codeword là tâm của cụm. Một codebook bao gồm K codeword của cùng một ROI và từ dữ liệu huấn luyện, ta được 5×5 codebook. Cuối cùng, ta đối chiếu mỗi vector SIFT của mỗi đoạn ở mỗi ROI với codebook tương ứng, sử dụng biểu đồ tần số của các codeword khác nhau và dùng biểu đồ này làm đặc trưng của ROI, sau đó ta nối dài 5×5 biểu đồ để thu được một vector biểu diễn ảnh khuôn mặt. Sử dụng SVM tuyến tính để huấn luyện biểu đồ của từng người.

Ở bước kiểm tra, ta cũng chia ảnh thành 5×5 khối, thu được 5×5 biểu đồ codeword sử dụng codebook đã huấn luyện. Nối dài biểu đồ này thu được vector biểu diễn ảnh, ta phân loại ảnh bằng phân loại SVM với mô hình huấn luyện.

2.1.3. Kết Quả Thực Nghiệm

Nhóm tác giả [2] sử dụng bộ dữ liệu AR [5] và XM2VTS để thực nghiệm.

Hình 7 Ảnh trong bộ dữ liệu AR.

Ảnh vào được nét xuống kích thước 270×230, nhóm thực nghiệm trên bộ dữ liệu AR với 119 đối tượng, huấn luyện trong bộ AR01, sủ dụng bộ AR02 – AR08, AR11, AR15 - AR21 và AR24 để kiểm tra. Kết quả thực nghiệm thu được ở Hình 8.

Recognition results (%)																
Facial expressions							Illumination						Occlusions			
AR02	AR03	AR04	AR15	AR16	AR17	AR05	AR06	AR07	AR18	AR19	AR20	AR08	AR11	AR21	AR24
100	**100**	**95.80**	**97.48**	**97.48**	**77.31**	**100**	**100**	**98.32**	**99.16**	**93.28**	**80.67**	**94.96**	**99.16**	**77.31**	**89.92**

Hình 8 Kết quả thực nghiệm trên bộ dữ liệu AR với 3 trạng thái: Cảm xúc khuôn mặt (Facial expressions), Ảnh sáng (Illumination), Che khuất (Occlusions).

2.1.4. Ưu và Nhược Điểm của Thuật Toán

2.1.4.1. Ưu Điểm

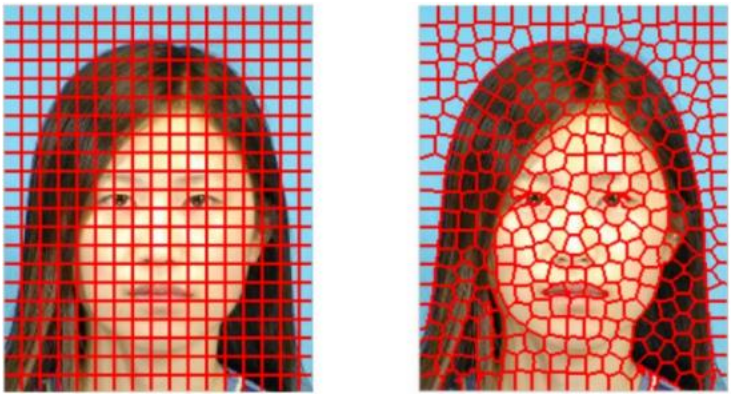
Thuật toán cho kết quả nhận dạng cao dưới nhiều điều kiện của ảnh, kể cả khuôn mặt có biểu đạt cảm xúc hay bị che khuất một phần mà chỉ cần huấn luyện bộ ảnh thường (giống như ảnh AR01).

2.1.4.2. Khuyết Điểm

Biểu diễn Bag of Word này chỉ hiệu quả khi ảnh không bị che khuất quá nhiều vì nếu không thì biểu đồ biểu diễn ảnh ở vùng một phần sẽ khác với biểu đồ cũng của vùng đó nhưng ở toàn phần. Vì lý do này nên Bag of Word không hiệu quả khi nhận dạng một phần mặt. [6]

2.1.5. 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]



Hình 9 Chia ảnh theo lưới vuông (trái) và chia ảnh theo superpixel (phải).

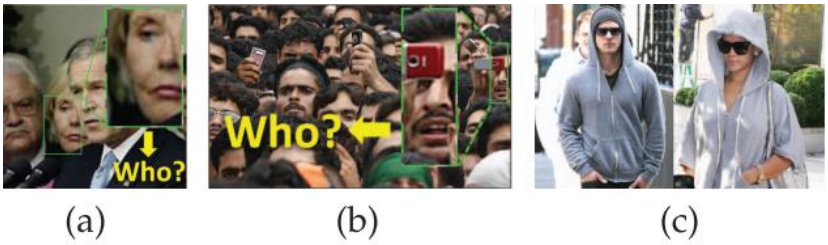
Ta xem mỗi vùng superpixel như là một ROI, sau đó sử dụng Bag of Word để huấn luyện và phân loại khuôn mặt.

2.2. Nhận Dạng Một Phần Khuôn Mặt Không Cần Canh Chỉnh

2.2.1. 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)

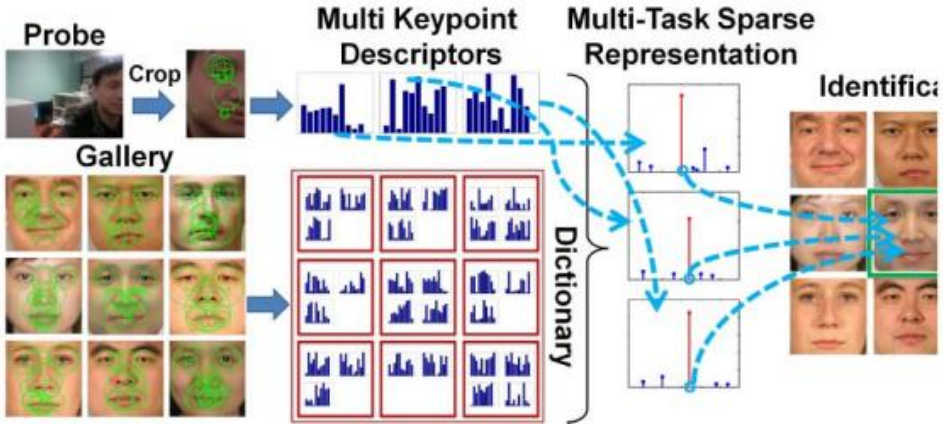
hay địa phương (Gabor, LBP) thường canh chỉnh và biểu diễn các ảnh khuôn mặt theo cùng một kích cỡ, sau đó nối dài các giá trị điểm ảnh hoặc trích xuất các vector đặc trưng theo số chiều nhất định. Tuy nhiên, với ảnh một phần khuôn mặt thì cách làm này không khả thi do mặt không đủ chi tiết cũng như 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. Do đó nhóm tác giả [6] đề xuất 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.



Hình 10 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 kính, áo khoác.

Nhóm đã đề xuất một 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. Nhóm tác giả đã phát triển 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 11 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. Nhóm tác giả gọi thuật toán này là MKD-SRC.

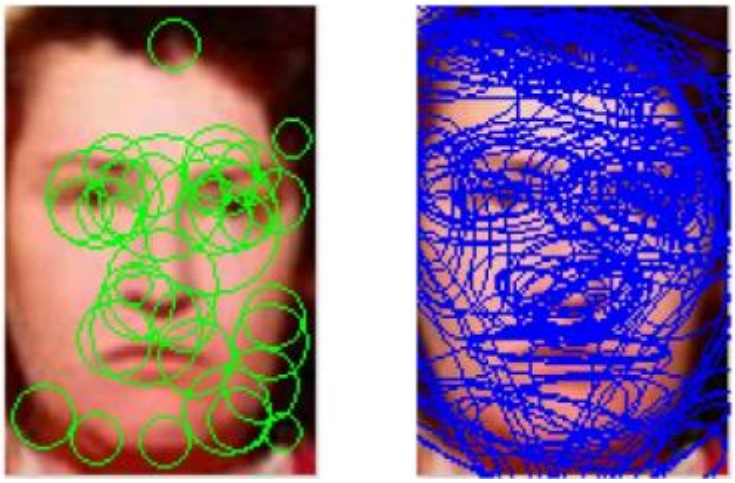


Hình 11 Mô tả ý tưởng cho thuật toán nhận dạng một phần khuôn mặt.

2.2.2. Chi Tiết Thuật Toán

2.2.2.1. Mô Tả Đa Điểm Chính (MKD)

Để xác định các điểm chính thì phép xác định SIFT do Lowe đề xuất [9] được sử dụng phổ biến do các điểm chính khi sử dụng phép này không bị ảnh hưởng khi ta xoay hay thay đổi kích thước ảnh. Tuy nhiên, các vùng điểm chính SIFT không bất biến dưới phép biến đổi affine và số lượng điểm chính bị giới hạn do SIFT chỉ tìm các cấu trúc có những tính chất tương đối đặc biệt. Do đa số khuôn mặt tương đối giống nhau nên ta cần phải xác định nhiều điểm chính. Nhóm tác giả [6] sử dụng cạnh Canny [10] có tỉ lệ bất biến do bài báo [11] đề xuất và kỹ thuật lắp hình bất biến affine do bài báo [12] đề xuất, từ đó xây dựng phép xác định điểm chính CanAff xác định nhiều điểm chính hơn phép xác định SIFT do trên khuôn mặt có các cạnh nhiều hơn các điểm đặc biệt (xem Hình 12).



Hình 12 So sánh cách xác định các điểm chính bằng SIFT và CanAff. Ảnh trái: Dùng SIFT xác định được 37 điểm chính. Ảnh phải: Dùng CanAff xác định 571 điểm chính (ảnh chỉ hiển thị 150 điểm chính đầu tiên).

Phép xác định CanAff đầu tiên trích xuất các cạnh với phép xác định cạnh Canny đa mức, sau đó với mỗi điểm cạnh, xác định láng giềng địa phương có tỉ lệ bất biến. Ta xác định kích thước đặc trưng của láng giềng địa phương bằng cách tìm cực trị địa phương dựa trên toán tử LoG chuẩn hóa tỉ lệ [13]. Tiếp theo, ta lấy vùng địa phương đã xác định đó biến thành hình bất biến affine không đồng hướng như trong bài báo [12], với điểm chính láng giềng được biến đổi một cách có lặp lại sử dụng thông tin từ ma trận moment thứ 2. Ở bước hội tụ, mỗi vùng affine bất biến có dạng giống hình ellipse

$$x^T M x = 1 \tag{1}$$

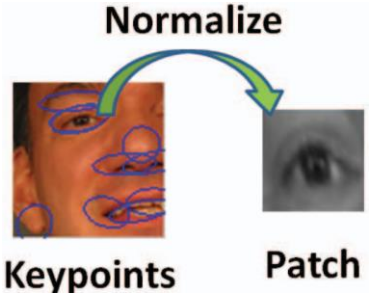
với các tham số ellipse

$$M = \begin{pmatrix} a & b \\ b & c \end{pmatrix} \tag{2}$$

xác định từ vùng láng giềng biến đổi affine, ta chuẩn hóa hình học thành hình tròn qua phép biến đổi affine

$$x' = M^{1/2} x \tag{3}$$

và vùng thu được được nén lại còn kích thước 40×40 điểm ảnh.



Hình 13 Chuẩn hóa vùng điểm chính mắt có dạng hình ellipse (ảnh trái) thành hình tròn (ảnh phải).

2.2.2.2. Mô Tả Mẫu Tam Phân Gabor (GTP)

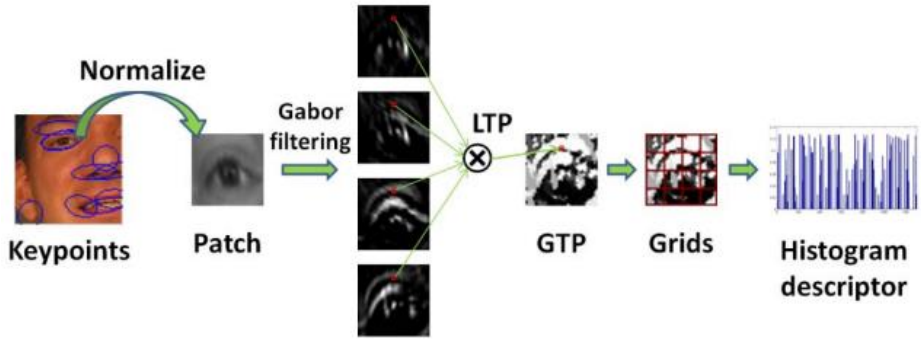
Khi vùng được xác định đã chuẩn hóa thành một kích thước cố định, ta xây dựng phép mô tả địa phương cho từng vùng như sau. Đầu tiên, ta sử dụng bộ lọc Gabor vào mỗi phần của ảnh do bộ lọc này phù hợp với cấu trúc ảnh địa phương. Nhân Gabor được định nghĩa như sau:

$$\psi_{\mu,\nu}(x,y) = \frac{\|k_{\mu,\nu}\|^2}{\sigma^2}\exp\left(-\frac{\|k_{\mu,\nu}\|^2\|z\|^2}{2\sigma^2}\right) \times \left[\exp\left(ik_{\mu,\nu}^T z\right) - \exp\left(-\frac{\sigma^2}{2}\right)\right] \quad (4)$$

với μ và ν lần lượt định nghĩa là hướng và tỉ lệ của nhân Gabor. $z = (x,y)^T$ và vector sóng $k_{\mu,\nu}$ được định nghĩa là

$$k_{\mu,\nu} = \left(k_\nu \cos\phi_\nu, k_\nu \sin\phi_\mu\right)^T \quad (5)$$

với $k_\nu = k_{\max}/f^\nu$, $k_{\max} = \pi/2$, $f = \sqrt{2}$ và $\phi_\mu = \pi\mu/8$. Do kích thước vùng ta xử lý tương đối nhỏ (40×40 điểm ảnh), nhân Gabor tại tỉ lệ đơn ($\nu = 0$) và bốn hướng ($\mu \in \{0,2,4,6\}$ tương ứng với 0^o, 45^o, 90^o và 135^o) với $\sigma = 1$. Hơn nữa, ta chỉ sử dụng nhân Gabor lẻ (phần ảo) do nhân này nhạy với các cạnh và vị trí của cạnh đó. Bốn nhân Gabor này có thể phân biệt được các chi tiết địa phương trong ảnh khuôn mặt. Hình 14 cho thấy 4 bộ lọc Gabor áp dụng vào các vùng địa phương, làm nổi bật các cạnh theo 4 hướng khác nhau (0^o, 45^o, 90^o và 135^o).



Hình 14 Các thành phần chủ yếu của phép Mô Tả GTP.

Với mỗi điểm ảnh (x,y) của vùng điểm chính đã chuẩn hóa, ta được 4 bộ lọc Gabor trả về kết quả như sau

$$f_i(x,y) = G_i(x,y) * I(x,y), \quad i = 0,1,2,3 \quad (6)$$

với $G_i = imag(\psi_{2i,0})$ là nhân Gabor lẻ thứ i và $*$ là toán tử chập. Tổ hợp 4 kết quả trả về sau khi sử dụng 4 bộ lọc trên thu được Mẫu Tam Phân Gabor [14]

$$GTP_t(x,y) = \sum_{i=0}^{3} 3^i[(f_i(x,y) < -t) + 2(f_i(x,y) > t)] \quad (7)$$

với t là ngưỡng dương nhỏ (nhóm tác giả chọn $t = 0.03$), vậy ta có tổng cộng $3^4 = 81$ mẫu tam phân khác nhau. Hình 14 cho thấy 4 điểm ảnh tương ứng nằm trong cùng một vị trí ở 4 ảnh kết quả sau khi lọc Gabor, tạo thành mẫu GTP.

Tiếp theo, chia 40×40 vùng thành $4 \times 4 = 16$ ô lưới nhỏ, mỗi ô có kích thước 10×10 điểm ảnh. Lập biểu đồ GTP trên mỗi ô lưới, sau đó nối dài các biểu đồ, thu được vector đặc trưng 1296 chiều ($4 \times 4 \times 81$). Để làm giảm sự ảnh hưởng của các giá trị ngoại lai, ta chuẩn hóa vector đặc trưng thành độ dài đơn vị, sau đó sử dụng hàm sigmoid $\tanh(ax)$, với a là hằng số để khử các giá trị này (nhóm tác giả chọn $a = 20$). Cuối cùng, sử dụng PCA để làm giảm số chiều của vector đặc trưng xuống còn M (nhóm tác giả chọn $M = 128$).

2.2.2.3. MKD-SRC

2.2.2.3.1. Xây Dựng Thư Viện Từ Điển

Biểu diễn mỗi ảnh bằng MKD, giả sử trong bộ dữ liệu một người có C lớp ảnh, ở lớp ảnh thứ c xác định được k_c điểm chính $p_{c_1}, p_{c_2}, \dots, p_{c_{k_c}}$, sử dụng mô tả GTP, ta thu được các mẫu tam phân $d_{c_1}, d_{c_2}, \dots, d_{c_{k_c}}$ với mỗi d_{c_i} là vector M chiều. Đặt

$$D_c = \left(d_{c_1}, d_{c_2}, \dots, d_{c_{k_c}}\right) \quad (8)$$

là tập các mẫu tam phân ở ảnh c, khi đó D_c là một từ điển con biểu diễn cho lớp c có kích thước $M \times k_c$. Từ đó, ta xây dựng thư viện từ điển cho bộ dữ liệu cho cả C lớp

$$D = (D_1, D_2, \dots, D_C) \quad (9)$$

lưu ý rằng D có tổng cộng $K = \sum_{c=1}^{C} k_c$ điểm chính, khi đó kích thước từ điển là $M \times K$.

Thông thường kích thước của K rất lớn, khoảng 1 triệu, do đó ta cần biểu diễn bất kỳ điểm chính từ ảnh kiểm tra dưới dạng tổ hợp tuyến tính thưa từ từ điển D.

2.2.2.3.2. Biểu Diễn Đa Nhiệm Thưa

Cho ảnh kiểm tra có n điểm chính

$$Y = (y_1, y_2, \dots, y_n) \quad (10)$$

khi đó nhóm tác giả [6] giải bài toán cực tiểu l_1

$$\hat{x}_i = \arg\min_{x_i} \|x_i\|_1 \quad (11)$$

sao cho

$$y_i = Dx_i, \;\; i = 1,2, \dots, n \quad (12)$$

với $X = (x_1, x_2, \dots, x_n) \in \mathbb{R}^{K \times n}$ là ma trận hệ số thưa và $\|\cdot\|_1$ là ký hiệu chuẩn l_1 của vector, tức $\|x\|_1 = \sum_i |x_i|$. Đây là bài toán đa nhiệm do X và Y có nhiều cột. Có nhiều thuật toán cực tiểu l_1 nhanh sử dụng để giải phương trình (11), ví dụ như thuật toán Đồng luân l_1 [15].

Lấy ý tưởng từ bài báo [8], nhóm chấp nhận SRC đa nhiệm dưới đây để xác định tính đồng nhất của ảnh kiểm tra.

$$\min_{c} r_c(Y) = \frac{1}{n}\sum_{i=1}^{n} \|y_i - D_c\delta(\hat{x}_i)\|_2^2 \qquad (13)$$

với $\delta_c(\cdot)$ là hàm số lựa chọn hệ số dựa theo lớp c. Phương trình (13) áp dụng tổng kết hợp giữa tái xây dựng phần còn lại của n điểm chính theo từng lớp và xác định tính đồng nhất dựa trên phần dư cuối cùng. Do đó, ta có thể nhận dạng ảnh kiểm tra bằng cách tính phương trình (11) và (13). Đây là thuật toán MKD-SRC, nhận dang khuôn mặt không cần canh chỉnh mặt.

2.2.2.3.3. Lọc Nhanh

Trong thực tế, kích thước (K) của từ điển D có thể đến hàng triệu, khiến việc giải phương trình (11) trở nên khó khăn. Do đó, nhóm sử dụng cách xấp xỉ nhanh sau đây, với mỗi điểm chính y_i trng ảnh kiểm tra, đầu tiên tính hệ số tương quan tuyến tính giữa y_i và tất cả điểm chính trong từ điển D.

$$c_i = D^T y_i, \;\; i = 1,2,\dots,n \qquad (14)$$

Khi đó với mỗi y_i, ta giữ L điểm chính ứng với L giá trị c_i lớn nhất ($L \ll K$), đưa đến bộ từ điển con nhỏ $D_{M\times L}^{(i)}$. Tiếp theo, thay D bằng $D^{(i)}$ trong phương trình (11), và phương trình (13) được thay đổi phù hợp.

2.2.2.4. Mã Giả Thuật Toán MKD-SRC

Thuật toán 1. Thuật toán MKD-SRC

Input: Thư viện ảnh gồm C lớp; ảnh kiểm tra I; tham số L

Output: Đồng nhất c với ảnh kiểm tra I

Quy trình: Trích xuất Mô Tả Đa Điểm Chính (GTP) từ mỗi ảnh thư viện và xây dựng từ điển $D = (D_1, D_2, \dots, D_C) \in \mathbb{R}^{M\times K}$

Nhận dạng:

Trích xuất MKD từ ảnh kiểm tra; $Y = (y_1, y_2, \dots, y_n) \in \mathbb{R}^{M\times n}$;

for $i = 1$ to n do

  Tính L điểm chính cao nhất từ phương trình (14), thu được thư viện con $D_{M\times L}^{(i)}$;

  Giải phương trình (11) với $D_{M\times L}^{(i)}$;

end

Giải phương trình (13) để xác định hình c đồng nhất với hình kiểm tra

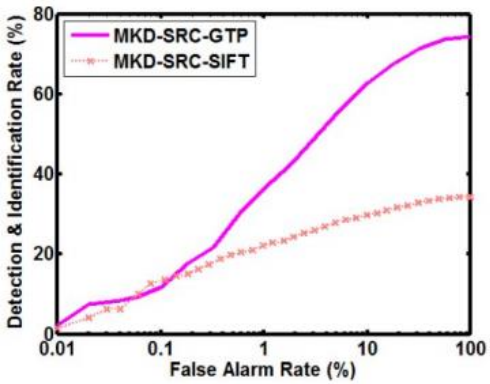
2.2.3. Kết Quả Thực Nghiệm

Nhóm tác giả [6] thực nghiệm trên 4 bộ dữ liệu, trong bài báo cáo này em trình bày trên 2 bộ: FRGCv2.0, và AR.

2.2.3.1. Nhận Dạng Một Phần Khuôn Mặt với Phần Mặt Tùy Ý

Hình 15 Ví dụ về ảnh khuôn mặt dùng trong thực nghiệm với một phần mặt: (a) Ảnh thư viện từ bộ dữ liệu FRGCv2.0 (b) ảnh một phần mặt từ bộ dữ liệu FRGCv2.0 dùng để kiểm tra.

Nhóm tác giả tạo ra các ảnh một phần khuôn mặt từ 16028 ảnh chính diện của 466 đối tượng từ bộ dữ liệu FRGCv2.0. Ảnh thư viện được chỉnh xuống kích thước 128×128 điểm ảnh dựa trên tọa độ 2 mắt. Hình 15 (a) là ảnh trong thư viện, từ cac ảnh này, nhóm tác giả tạo ra ảnh phần mặt tùy ý bằng cách xoay ảnh ngẫu nhiên dựa theo phân phối Gauss với trung bình 0 và độ lệch chuẩn 10^o, sau đó chọn ngẫu nhiên một vị trí với kích thước ngẫu nhiên từ đó trích xuất ra một phần mặt. Cuối cùng, ảnh một phần mặt được canh chỉnh thành kích thước $h \times w$, với h và w là phân phối đều trong đoạn giá trị $[64,256]$, thu được ảnh như Hình 15 (b). Sử dụng thuật toán MKD-SRC-GTP và MKD-SRC-SIFT ta có đường cong ROC như Hình 16, cho thấy thuật toán MKD-SRC-GTP tốt hơn MKD-SRC-SIFT, nhưng nhìn chung vẫn chưa giải quyết ổn bài toán nhận dạng.



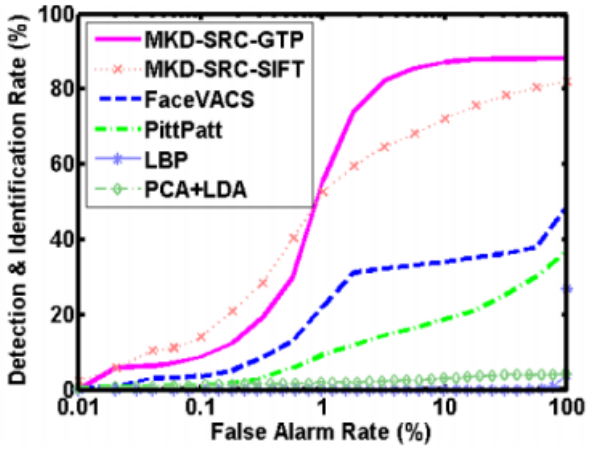
Hình 16 Đường cong ROC khi nhận dạng khuôn mặt với một phần mặt tùy ý, sử dụng bộ dữ liệu FRGCv2.0.

2.2.3.2. Nhận Dạng Ảnh Chính Diện bị Che

Sử dụng bộ dữ liệu AR với 10135 đối tượng và 10135 ảnh làm thư viện (1 đối tượng / 1 ảnh), nhóm tác giả thực nghiệm trên 2 tập kiểm tra, một tập $\mathcal{P}_\mathcal{G}$ chứa 1530 ảnh (gồm ảnh không bị che và ảnh chính diện) trong bộ AR thuộc 10135 đối tượng (khác với ảnh thư viện), tập còn lại $\mathcal{P}_\mathcal{N}$ chứa 10000 ảnh từ các đối tượng khác 10135 đối tượng trên. Tất cả ảnh được chỉnh kích thước còn 128×128 điểm ảnh.

Hình 17 Ảnh trong bộ dữ liệu AR. Dòng trên: Ảnh thư viện. Dòng dưới: ảnh kiểm tra.

Hình 18 cho thấy thuật toán MKD-SRC-SIFT và MKD-SRC-GTP có kết quả tốt hơn các thuật toán nhận dạng khuôn mặt còn lại. Với tốc độ nhận dạng sai (False Alarm Rate) 1%, thuật toán MKD-SRC-GTP có thể loại bỏ 99% ảnh $\mathcal{P}_{\mathcal{N}}$ và xác định hơn 55% ảnh kiểm tra $\mathcal{P}_{\mathcal{G}}$.



Hình 18 Đường cong ROC khi nhận dạng ảnh chính diện bị che, sử dụng bộ dữ liệu AR.

Nhìn chung đây là vấn đề khó do toàn bộ ảnh trong thư viện là ảnh chính diện không bị che khuất còn ảnh kiểm tra $\mathcal{P}_{\mathcal{G}}$ có bị che, đôi khi có ảnh bị phơi sang còn ảnh trong $\mathcal{P}_{\mathcal{N}}$ không bị che, và tập thư viện chỉ có 1 ảnh cho 1 người.

2.2.4. Ưu và Nhược Điểm của Thuật Toán

2.2.4.1. Ưu Điểm

Sử dụng thuật toán này cho ra kết quả nhận dạng một phần khuôn mặt tốt hơn các phương pháp phổ biến trước đây như PCA và LDA, LBP. Ngoài ra, thuật toán đề xuất một phương pháp xác định các điểm chính MKD cho ảnh khuôn mặt không chính diện và xây dựng các điểm chính phù hợp trong ảnh bằng phép phân loại biểu diễn thưa (SRC). [16]

2.2.4.2. Nhược điểm

Do thuật toán này sử dụng quá nhiều điểm chính cũng như kích thước từ điển lớn nên dẫn đến chi phí tính toán cao. Ngoài ra, thuật toán này không chú ý đến thông tin hình học của tập đặc trưng [17].

2.2.5. 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, 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.

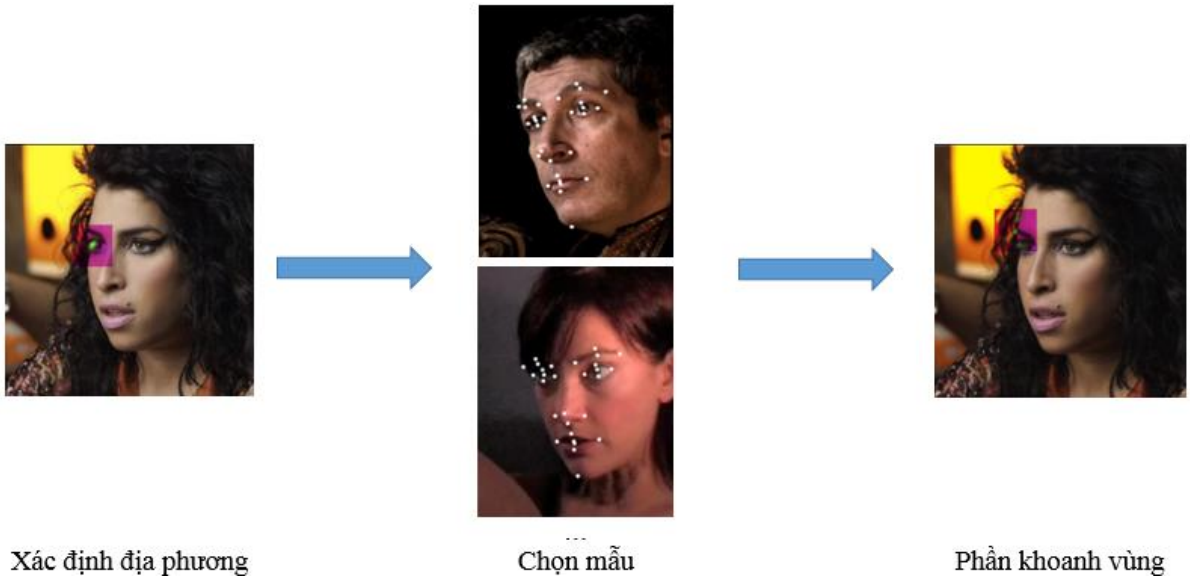
2.3. Khoanh Vùng Một Phần Khuôn Mặt Sử Dụng Các Mẫu Đồng Dạng

2.3.1. 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. Nhóm tác giả [18] đề xuất một thuật toán khoanh vùng các điểm chính đã được định sẵn trước dưới nhiều điều kiện ảnh khác nhau. Trong bài báo [18], nhóm tác giả này 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.

Hình 19 Ảnh kết quả sau khi khoanh vùng của nhóm tác giả *[18]*.

Nhóm tác giả xem việc khoanh vùng phần mặt 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, nhóm tác giả 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 [19]. 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), nhóm tác giả sử dụng SVM với hàm nhân bán kính cơ sở [20].

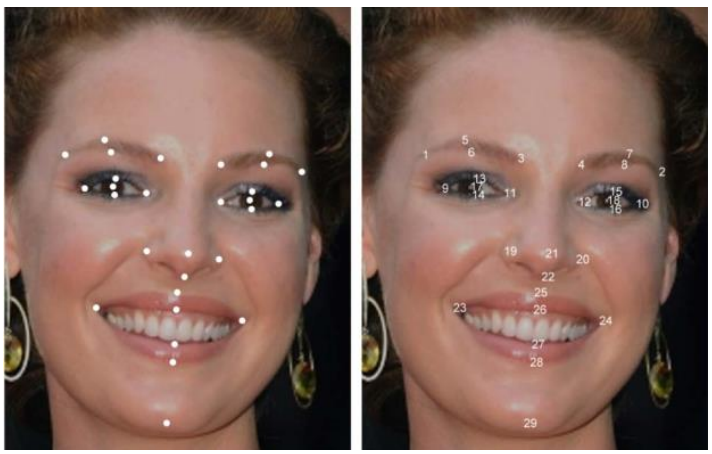


Hình 20 Tóm tắt thuật toán của nhóm tác giả *[18]*. Ảnh trái: Sử dụng xác định địa phương để xác định các điểm chính (ví dụ: cạnh mắt). Ảnh giữa: Chọn mẫu từ tập huấn luyện. Ảnh phải: Kết quả khoanh vùng điểm chính.

2.3.2. Chi Tiết Thuật Toán

2.3.2.1. Xây Dựng Bộ Ảnh Huấn Luyện

Nhóm tác giả xây dựng bộ dữ liệu LFPW gồm 3000 ảnh mặt được tải trên mạng, mỗi ảnh có 35 điểm chính do các công nhân từ công ty Amazon Mechanical Turk (MTurk) thực hiện, mỗi điểm ảnh do 3 công nhân thực hiện, lấy kết quả trung bình làm điểm chính. Hình 21 mô tả vị trí của thể của 35 điểm này, đây là bộ huấn luyện dùng làm dữ liệu thật. Với ảnh trong bộ huấn luyện, nhóm tác giả chỉnh kích thước ảnh sao cho khoảng cách giữa 2 đồng tử khoảng 55 điểm ảnh.



Hình 21 Ảnh trong bộ dữ liệu LFPW với 35 điểm chính trên mặt do công nhân công ty MTurk thực hiện thủ công. Trong bộ dữ liệu này có đánh số thứ tự quy ước vị trí các điểm chính tương ứng với vị tri trên mặt (ảnh phải).

2.3.2.2. Xác Định Địa Phương

Nhóm tác giả xây dựng một cửa sổ trượt trượt qua vùng khuôn mặt, cửa sổ trượt này là phân loại SVM kết hợp mô tả SIFT xám làm đặc trưng [9]. Sau đó, tính cửa sổ đặc trưng SIFT trên 2 tỉ lệ: $1/4$ và $1/2$ khoảng cách đồng tử. Nối dài hai mô tả SIFT này để tạo thành vector đặc trưng 256 chiều dùng cho phân loại SVM. Sau đó, bộ mô tả trả về điểm tại mỗi điểm x trong

ảnh, tương ứng với khoảng cách từ điểm đó đến siêu phẳng dùng trong phân loại. Điểm mô tả $d(x)$ có ý nghĩa khả năng điểm chính mong muốn nằm tại điểm x trong ảnh, do đó ta chuẩn hóa về xác suất bằng cách chia cho tổng các điểm trong cửa sổ xác định, được điểm $P(x|d)$, tức xác suất điểm mong muốn nằm tại x khi đã biết tất cả điểm trong cửa sổ xác định.

Tuy nhiên, do xác định địa phương chưa đủ tốt nên vị trí đúng của điểm chính chưa chắc sẽ nằm ngay điểm xác định cao nhất do một số lý do như khuôn mặt bị tóc che một phần, có đeo kính, đeo tai nghe, … Ở phần sau, nhóm tác giả trình bày cách xây dựng xác định toàn cục để khắc phục trường hợp lỗi ở trên.

2.3.2.3. Xác Định Toàn Cục

Mặt dù ảnh khuôn mặt có nhiều dạng khác nhau tùy vào góc chụp nhưng cấu trúc hình học cũng như kết cấu khuôn mặt chi phối đến bố cục các phần trong khuôn mặt và vị trí trong ảnh. Nhóm tác giả xét tất cả các phần trong khuôn mặt để phát triển thuật toán xác định toàn cục để tìm ra các điểm chính.

Đặt

$$X = \{x^1, x^2, \dots, x^n\} \tag{15}$$

là vị trí chính xác của n điểm chính, với x^i là vị trí của điểm chính thứ i. Đặt

$$D = \{d^1, d^2, \dots, d^n\} \tag{16}$$

là điểm với vị trí xác định tương ứng, với d^i kết quả điểm tại vị trí xác định địa phương thứ i. Ta muốn tìm giá trị X sau cho tối đa hóa xác suất của X khi biết điểm từ các xác định toàn cục, tức

$$X^* = \arg\max_X P(X|D) \tag{17}$$

Đặt X_k, với $k = 1, \dots, m$ là vị trí của n điểm chính ở vị trí thứ k của m mẫu trong bộ huấn luyện, đặt $X_{k,t}$ là vị trí điểm chính trong mẫu k đã được biến đổi theo phép biến đổi t, ta gọi $X_{k,t}$ là mẫu toàn cục. Nếu ta giả sử mỗi X được tạo từ một trong các mẫu toàn cục $X_{k,t}$, ta có thể khai triển $P(X|D)$ như sau

$$P(X|D) = \sum_{k=1}^{m} \int_{t \in T} P\left(X \middle| X_{k,t}, D\right) P\left(X_{k,t} \middle| D\right) dt \tag{18}$$

lúc này ta đã thêm tập m mẫu X_k với phép biến đổi t tương ứng. Ta có thể xem điểm chính tại x^i độc lập có điều kiện với các điểm chinh khác khi biết $X_{k,t}$, ta viết điểm chính này là $x_{k,t}^i$, tức vị trí của x^i đã biến đổi theo phép biến đổi t được điểm chính tương ứng trong mẫu k. Ta viết lại biểu thức đầu tiên của (18) thành

$$P\left(X \middle| X_{k,t}, D\right) = \prod_{i=1}^{n} P\left(x^i \middle| x_{k,t}^i, d^i\right) \tag{19}$$

$$= \prod_{i=1}^{n} \frac{P\left(x_{k,t}^i \middle| x^i, d^i\right) P\left(x^i \middle| d^i\right)}{P\left(x_{k,t}^i \middle| d^i\right)} \tag{20}$$

Do việc biết vị trí chính xác của điểm chính có thể giúp ta biết thông tin từ cửa sổ xác định rằng mẫu nào cũng như phép biến đổi nào đã được sử dụng để tạo ra ảnh, nên

$$P\left(x_{k,t}^{i}\middle|x^{i},d^{i}\right)=P\left(x_{k,t}^{i}\middle|x^{i}\right) \tag{21}$$

đồng thời do mối quan hệ giữa mô hình biến đổi điểm chính và giá trị chính xác của điểm chính thường bất biến nên đặt $\Delta x_{k,t}^{i}=x_{k,t}^{i}-x^{i}$. Với nhận xét này, ta viết lại (20) thành

$$P\left(X\middle|X_{k,t},D\right)=\prod_{i=1}^{n}\frac{P\left(\Delta x_{k,t}^{i}\right)P\left(x^{i}\middle|d^{i}\right)}{P\left(x_{k,t}^{i}\middle|d^{i}\right)} \tag{22}$$

Sử dụng định lý Bayes để xử lý biểu thức thứ hai của phương trình (18), thu được

$$P\left(X_{k,t}\middle|D\right)=\frac{P\left(D\middle|X_{k,t}\right)P\left(X_{k,t}\right)}{P(D)} \tag{23}$$

$$=\frac{P\left(X_{k,t}\right)}{P(D)}\prod_{i=1}^{n}P\left(d^{i}\middle|x_{k,t}^{i}\right) \tag{24}$$

với các giá trị d^{i} độc lập có điều kiện với nhau.

Áp dụng định lý Bayes, ta viết lại (24) thành

$$P\left(X_{k,t}\middle|D\right)=\left[\frac{P\left(X_{k,t}\right)}{P(D)}\frac{\prod_{i=1}^{n}P\left(d^{i}\right)}{\prod_{i=1}^{n}P\left(x_{k,t}^{i}\right)}\right]\prod_{i=1}^{n}P\left(x_{k,t}^{i}\middle|d^{i}\right) \tag{25}$$

$$=C\prod_{i=1}^{n}P\left(x_{k,t}^{i}\middle|d^{i}\right) \tag{26}$$

do các biểu thức trong ngoặc vuông ở (25) giờ là hằng số nên ta đặt các biểu thức ấy là C. Kết hợp (17), (18), (22) và (26) ta được

$$X^{*}=\arg\max_{X}\sum_{k=1}^{m}\int_{t\in T}\prod_{i=1}^{n}P\left(\Delta x_{k,t}^{i}\right)P\left(x^{i}\middle|d^{i}\right)dt \tag{27}$$

Biểu thức $P\left(\Delta_{k,t}^{i}\right)$ là phân phối Gauss trên 2 chiều có tâm tại $x_{k,t}^{i}$, mỗi i có phân phối Gauss riêng. Phân phối này có ý nghĩa khả năng vị trí điểm chính trong mẫu toàn cục khớp với vị trí chính xác, do đó nếu ta dùng nhiều mẫu để xây dựng mẫu toàn cục, tức m rất lớn thì phân phối sẽ có phương sai thấp và khớp với kết quả.

Để ước lượng ma trận hiệp phương sai của vị trí điểm chính, ta làm như sau. Với mỗi mẫu X_j trong bộ huấn luyện, ta tìm mẫu X_k trong các mẫu còn lại và phép biến đổi t khớp L_2 nhất với X_j. Ta tính sai số $X_j - X_{k,t}$ và chuẩn hóa theo khoảng cách đồng tử. Sai số chuẩn hóa dùng để tính ma trận hiệp phương sai cho mỗi vị trí điểm chính.

Ta tính biểu thức $P\left(x^{i}\middle|d^{i}\right)$ bằng cách ước lượng vị trí x^{i} với điểm chính i và tìm giá trị trả về trong cửa sổ phát hiện thứ i, tức $d^{i}\left(x^{i}\right)$, sau đó chuẩn hóa về xác suất bằng cách chia cho tổng các $d^{i}(x)$ trong cửa sổ phát hiện.

2.3.2.4. Chọn Mẫu Toàn Cục $X_{k,t}$

Tính toán tổng các tích phân ở (27) rất phức tạp do ta phải tính tổng trên toàn bộ mô hình toàn cục k và trên cả phép biến đổi tương ứng t. Tuy nhiên, từ (18) ta thấy rằng nếu $P(X_{k,t}|D)$ rất nhỏ khi biết k và t thì biểu thức này có thể thay thế cho tổng và tích phân toàn cục. Do đó, nhóm tác giả [18] sử dụng tích phân Monte Carlo để xử lý mẫu toàn cục k với phép biến đổi t sao cho $P(X_{k,t}|D)$ lớn.

Nhóm đã sử dụng Mẫu Đồng Dạng Ngẫu Nhiên (RANdom SAmple Consensus - RANSAC) với thuật toán như sau

	Thuật toán 2. Chọn mẫu RANSAC
	input: r, D, m^*
	output: m^* cặp (k,t) có điểm cao nhất
1	Lặp lại r lần
2	Chọn ngẫu nhiên mẫu k.
3	Chọn ngẫu nhiên 2 điểm chính từ output của phát hiện địa phương $D = \{d^i\}$ trong ngăn chứa.
4	Tìm phép biến đổi t tương ứng nhằm canh mẫu sao cho khớp với 2 điểm chính này (Hình 22).
5	Đánh giá mức độ khớp cho mọi phần điểm chính i trên khuôn mặt với mỗi cặp (k,t) từ phương trình (26).
6	Thêm cặp (k,t) vào danh sách M các mẫu có thể sử dụng được xếp hạng bằng điểm.
7	Dừng lặp
8	Lấy m^* cặp (k,t) có điểm cao nhất từ M để xác định dạng toàn cục.

Khi thực nghiệm, nhóm tác giả chọn $r = 10000, m^* = 100$.

Hình 22 Canh khớp mẫu (2) với ảnh input với 2 điểm chính (1) như ảnh (3), ta được ảnh (4).

2.3.2.5. Ước Lượng X

Sau khi dùng RANSAC tìm ra danh sách M gồm m^* mẫu toàn cục $X_{k,t}$ có $P(X_{k,t}|D)$ lớn nhất, khi đó ta xấp xỉ bài toán tối ưu cho X trong phương trình (27) như sau

$$X^* = \arg\max_X \sum_{k,t \in M} \prod_{i=1}^{n} P(\Delta x_{k,t}^i) P(x^i | d^i) \qquad (28)$$

lúc này ta tính tổng dựa trên các giá trị $k, t \in M$. Để tìm giá trị lớn nhất X^*, đầu tiên ta cần ước lượng giá trị đầu x_0^i cho mỗi điểm chính i như sau

$$x_0^i = \arg\max_{x^i} \sum_{k,t \in M} P\left(\Delta x_{k,t}^i\right) P\left(x^i \middle| d^i\right) \tag{29}$$

công thức này tương đương với giải x_0^i bằng cách đặt mọi $P\left(\Delta x_{k,t}^j\right)$ và $P\left(x^j \middle| d^j\right)$ là hằng số với mọi $j \neq i$. Để tính từng x_0^i, ta chỉ đơn thuần nhân output xác định đã chuẩn hóa với hàm Gauss có tâm tại $x_{k,t}^i$, sau đó, ta tìm vị trí ảnh x_0^i với tổng của kết quả tích là lớn nhất. Ta có thể dùng giá trị ước lượng đầu $x_0^i, i = 1,2, \dots, n$ để tối ưu hóa (28) để tìm ước lượng x^{i*} cuối cùng, từ đó ra kết quả X^*.

Thuật toán 3. Ước lượng X

input: ds
output: ds

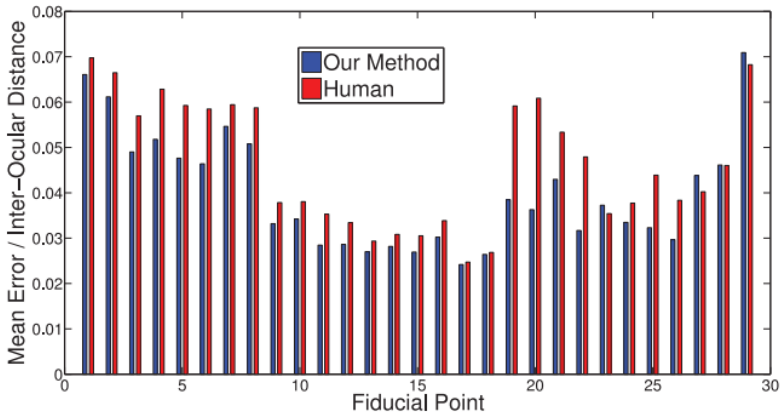
1 Với mỗi điểm chính i trên mặt:
2 Tính phân phối từ M mẫu đã canh chỉnh.
3 Với mỗi m^* mẫu trên cùng trong M (cặp (k,t)):
4 Nhân output của phát hiện địa phương đã chuẩn hóa với phân phối toàn cục của các điểm chính trong mẫu, thu được điểm số tại mỗi vị trí điểm ảnh.
5 Cộng tất cả điểm số để được điểm số cuối cùng tại mỗi vị trí điểm ảnh và chọn điểm lớn nhất.

2.3.3. Kết Quả Thực Nghiệm

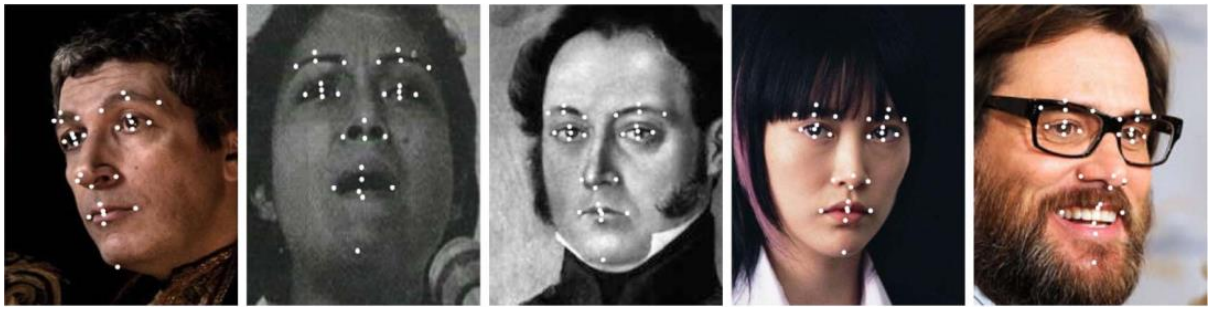
Nhóm tác giả tạo bộ dữ liệu LFPW đã có sẵn các điểm chính chuẩn do công nhân từ công ty MTurk thực hiện thủ công như Hình 21 (mỗi điểm có 3 công nhân đánh giá), sau đó thực nghiệm trên chính bộ dữ liệu này (không dùng điểm chính chuẩn), dùng 1100 ảnh huấn luyện và 300 ảnh kiểm tra. Ảnh huấn luyện dùng để huấn luyện bộ xác định dựa trên SVM và dùng để tính mô hình toàn cục X_k, thu được kết quả như Hình 23. Sau đó, nhóm đánh giá kết quả sai số từng vị trí so với điểm chính chuẩn bằng cách tính tỉ lệ khoảng cách trung bình điểm chính thu được so với kết quả của 3 công nhân MTurk và khoảng cách đồng tử (đối với điểm chính thu được), và tỉ lệ giữa trung bình 3 điểm chính của 3 công nhân và khoảng cách đồng tử (đối với điểm chính chuẩn). Hình 24 cho thấy kết quả so sánh thuật toán xác định điểm chính so với khoảng cách trung bình của 3 công nhân MTurk, ta thấy rằng khoảng cách này thường lớn hơn khoảng cách của thuật toán đến trung bình điểm được làm thủ công. Cần lưu ý rằng các điểm mắt (9-18) chính xác nhất, còn điểm mũi và miệng (19-29) khá tệ, và cằm và long mày (1-8, 29) kém nhất.

Hình 25 cho thấy một số điểm lỗi, ví dụ như hình thứ 2 và thứ 5 từ trái qua, điểm chính ở cằm không chính xác, lỗi này xảy ra khi miệng mở. Hình 26 là kết quả trên bộ dữ liệu LFW.

Hình 23 Ảnh thực nghiệm với bộ dữ liệu LFPW.



Hình 24 Sai số trung bình của kết quả xác định điểm chính so với trung bình phương sai của điểm chính chuẩn được làm thủ công trên bộ dữ liệu LFPW. Kết quả cho thấy thuật toán đưa ra chính xác hơn.



Hình 25 Ảnh kết quả xác định điểm chính trong bộ dữ liệu LFPW với một số điểm lỗi.

Hình 26 Ảnh kết quả từ bộ dữ liệu LFW với 55 điểm chính, sử dụng OpenCV để xác định khuôn mặt.

2.3.4. Ưu và Nhược Điểm của Thuật Toán

2.3.4.1. Ưu Điểm

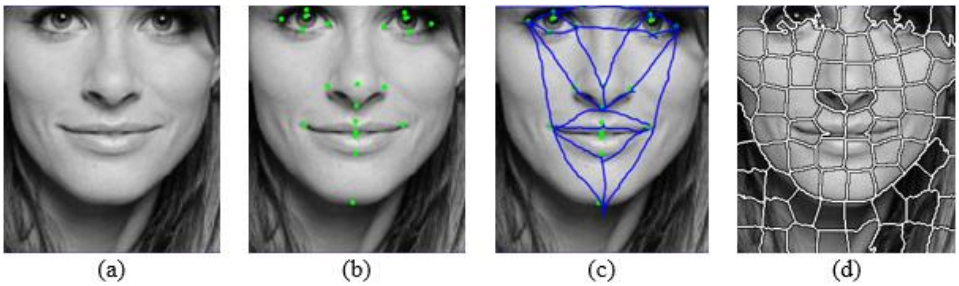
Thuật toán xác định chính xác nhiều điểm chính trên khuôn mặt, nhất là các điểm không có độ dốc cao và áp dụng trong nhiều điều kiện ảnh thực tế như ảnh chân dung, mặt cảm xúc, ảnh chiếu sang, chất lượng ảnh không cao.

2.3.4.2. Nhược điểm

Hạn chế của thuật toán này là việc sử dụng nhiều xác định địa phương cho các điểm chính (SVM + SIFT), do đó thuật toán mất tầm 10 giây để xác định 29 điểm chính trên toàn bộ ảnh (sử dụng Intel Core i7 3.06GHz), do đó, những phần mềm yêu cầu theo thời gian thực không thể sử dụng thuật toán này [21] [22].

2.3.5. 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. Xem Hình 27, 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].

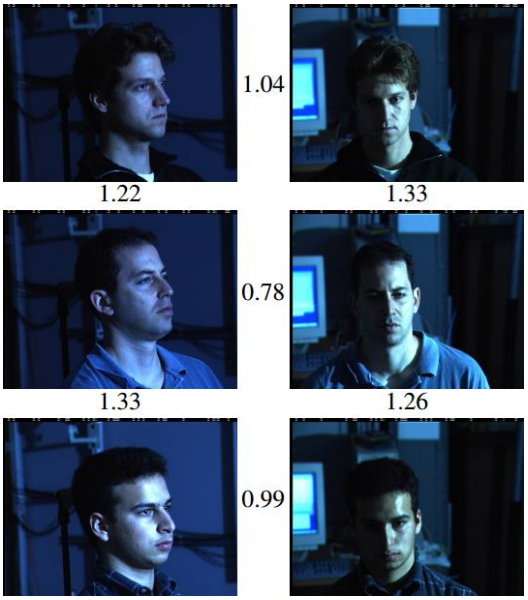


Hình 27 (a): Ảnh vào. (b) Xác định các điểm chính trên khuôn mặt. (c). Từ các điểm chính, chia thành các phần của khuôn mặt (d) Từ ảnh (a), lấy superpixel, với mỗi phần đã chia từ (c), chọn superpixels nằm trong phần đó.

2.4. Nhận Dạng Khuôn Mặt Sử Dụng Thuật Toán FaceNet

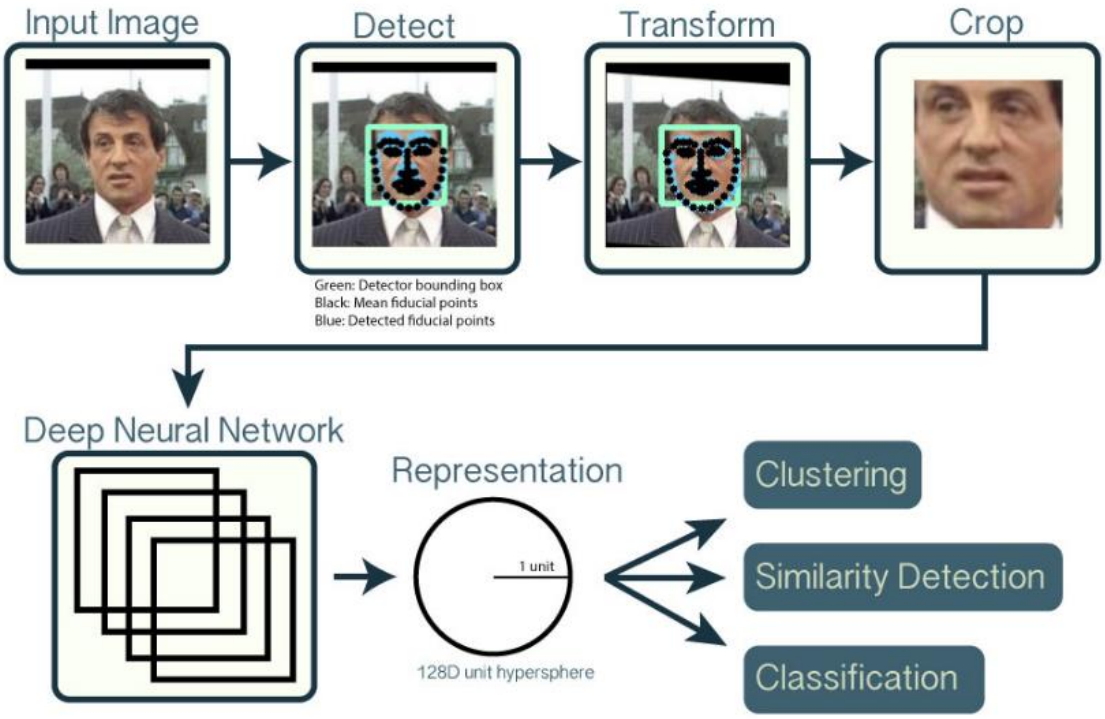
2.4.1. Tóm Tắt

Nhóm tác giả [23] 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 (xem Hình 28).



Hình 28 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.

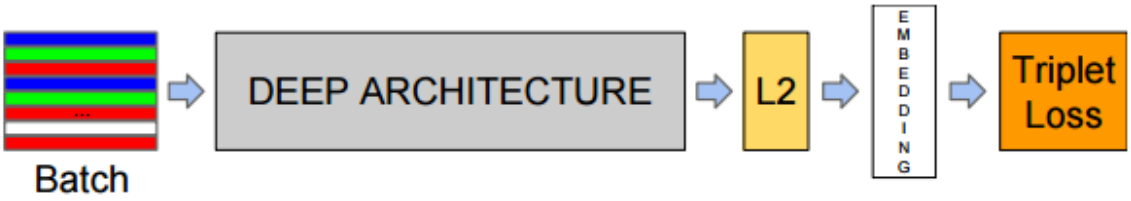
Sau khi thực hiện phép nhúng, thu được vector đặc trưng thì ta có thể thực hiện được 3 bài toán: Kiểm tra khuôn mặt, ta chỉ cần phân ngưỡng khoảng cách giữa 2 vector đặc trưng của 2 khuôn mặt. Nhận dạng khuôn mặt là bài toán phân loại k-NN. Phân cụm khuôn mặt sử dụng k-mean (xem tóm tắt cách nhận dạng khuôn mặt sử dụng FaceNet ở Hình 29).



Hình 29 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). [24]

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 [25], 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 [26], mạng còn lại sử dụng mô hình Inception từ GoogLeNet [27].

2.4.2. Chi Tiết Thuật Toán

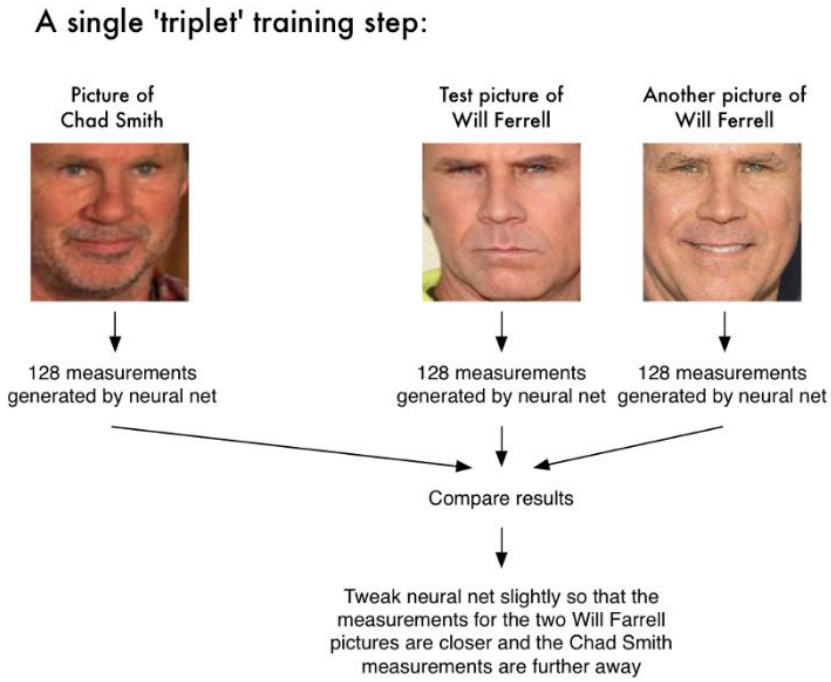


Hình 30 Cấu trúc mô hình, mạng bao gồm khối lớp vào (batch), đi qua cấu trúc Mạng Neuron Tích Chập Sâu (Convolution Neural Network – CNN), sau đó chuẩn hóa theo L_2 và đưa vào hệ thống nhúng. Trong quá trình huấn luyện có sử dụng bộ ba sai số.

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 $\mathbb{R}^d$ sao cho khoảng cách bình phương của tất cả khuôn mặt cùng loại phải nhỏ hơn khoảng cách bình phương với mặt khác loại.

2.4.2.1. Bộ Ba Sai Số

Ta biểu diễn phép nhúng là hàm $f(x) \in \mathbb{R}^d$ 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 x_i^a là ảnh kiểm tra người i, x_i^p là ảnh của người i trong bộ dữ liệu và x_i^n là ảnh không phải của người i (xem Hình 31), ta muốn rằng khoảng cách từ x_i^a đến x_i^p phải ngắn hơn khoảng cách từ x_i^a đến x_i^n, tức $d\left(x_i^a, x_i^p\right) < d(x_i^a, x_i^n)$.



Hình 31 Ví dụ về bộ ba sai số, ảnh bên trái (Chad Smith) là x_i^n, ảnh giữa (Will Ferrell) là x_i^a, ảnh phải (Will Ferrell) là x_i^p.

Do đó, ta muốn

$$\left\|f(x_i^a) - f\left(x_i^p\right)\right\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2, \tag{30}$$

$$\forall (f(x_i^a), f\left(x_i^p\right), f(x_i^n)) \in T \tag{31}$$

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

$$L = \sum_i^N \left[\left\|f(x_i^a) - f\left(x_i^p\right)\right\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha\right]_+ \tag{32}$$

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.

2.4.2.2. 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 x_i^a của người i, ta chọn x_i^p sao cho

$$\arg\max_{x_i^p} \left\|f(x_i^a) - f\left(x_i^p\right)\right\|_2^2 \tag{33}$$

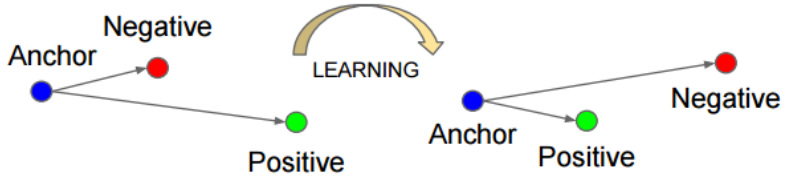
và chọn x_i^n sao cho

$$\arg\min_{x_i^n} \|f(x_i^a) - f(x_i^n)\|_2^2 \tag{34}$$

Điều này có nghĩa trong tập ảnh cùng đối tượng với x_i^a, ta chọn ảnh x_i^p (hard positive) sao cho khoảng cách giữa chúng là lớn nhất và trong tập ảnh khác đối tượng với x_i^a, chọn ảnh x_i^n (hard negative) sao cho khoảng cách giữa chúng là nhỏ nhất, khi đó có khả năng xảy ra trường hợp

$$\left\|f(x_i^a) - f\left(x_i^p\right)\right\|_2^2 > \|f(x_i^a) - f(x_i^n)\|_2^2 \tag{35}$$

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 x_i^a và x_i^p xuống thấp nhất và kéo dài khoảng cách giữa x_i^a và x_i^n ra xa nhất.



Hình 32 Bộ ba sai số tối thiểu hóa khoảng cách giữa ảnh vào (Anchor) và ảnh cùng loại với ảnh vào (Positive) và tối đa hóa khoảng cách giữa ảnh vào và ảnh khác loại với ảnh vào (Negative).

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 x_i^a và x_i^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 x_i^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 x_i^n sao cho

$$\left\|f(x_i^a) - f\left(x_i^p\right)\right\|_2^2 < \|f(x_i^a) - f(x_i^n)\|_2^2 \tag{36}$$

Ta gọi các mẫu x_i^n thỏa bất đẳng thức (36) là semi-hard do khoảng cách từ mẫu này đến x_i^a xa hơn khoảng cách đến x_i^p, nhưng ta vẫn gặp khó khăn do bình phương khoảng cách sát với khoảng cách x_i^a đến x_i^p. Các mẫu x_i^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)

$$\left\|f(x_i^a) - f\left(x_i^p\right)\right\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \tag{37}$$

sau đó lấy đạo hàm hàm L theo từng biến x_i^a, x_i^p, x_i^n

$$\frac{\partial L}{\partial x_i^a} = \sum_{i=1}^{N} \begin{cases} 2\left(f(x_i^n) - f\left(x_i^p\right)\right), & \text{nếu (37)} \geq 0 \\ 0, & \text{ngược lại} \end{cases} \quad (38)$$

$$\frac{\partial L}{\partial x_i^p} = \sum_{i=1}^{N} \begin{cases} -2\left(f(x_i^a) - f\left(x_i^p\right)\right), & \text{nếu (37)} \geq 0 \\ 0, & \text{ngược lại} \end{cases} \quad (39)$$

$$\frac{\partial L}{\partial x_i^n} = \sum_{i=1}^{N} \begin{cases} -2\left(f(x_i^a) - f(x_i^n)\right), & \text{nếu (37)} \geq 0 \\ 0, & \text{ngược lại} \end{cases} \quad (40)$$

sau đó, ta cập nhật tham số hàm sai số

$$f(x_i^a) = f(x_i^a) - \delta \frac{\partial L}{\partial x_i^a} \quad (41)$$

$$f\left(x_i^p\right) = f\left(x_i^p\right) - \beta \frac{\partial L}{\partial x_i^p} \quad (42)$$

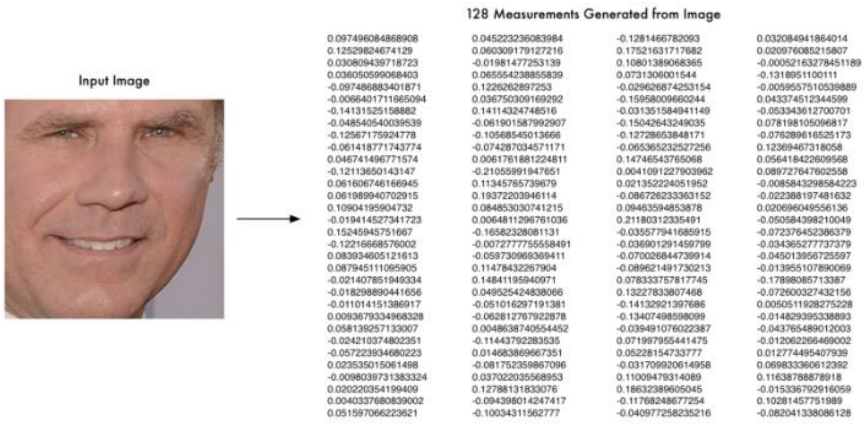
$$f(x_i^n) = f(x_i^n) - \gamma \frac{\partial L}{\partial x_i^n} \quad (43)$$

với δ, β, γ là tốc độ học.

Khi thực nghiệm, nhóm tác giả sử dụng khối gồm 1800 mẫu.

2.4.2.3. 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 $\delta = \beta = \gamma = 0.05$, 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.



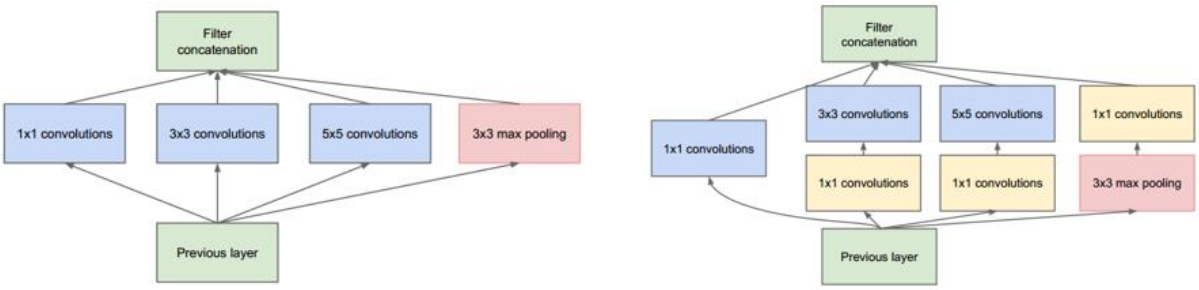
Hình 33 Ảnh vào (trái) sau khi huấn luyện, thu được vector 128 chiều (phải) [32].

Nhóm tác giả sử dụng 2 kiến trúc để học, một kiến trúc của Zeiler và Fergus [26] (xem Hình 34). Nhóm tác giả [23] thêm lớp tích chập $1 \times 1 \times d$, thu được mô hình sâu 22 lớp với 140 triệu tham số và cần 1.6 tỷ toán tử/giây/ảnh.

layer	size-in	size-out	kernel	param	FLPS
conv1	220×220×3	110×110×64	7×7×3, 2	9K	115M
pool1	110×110×64	55×55×64	3×3×64, 2	0	
rnorm1	55×55×64	55×55×64		0	
conv2a	55×55×64	55×55×64	1×1×64, 1	4K	13M
conv2	55×55×64	55×55×192	3×3×64, 1	111K	335M
rnorm2	55×55×192	55×55×192		0	
pool2	55×55×192	28×28×192	3×3×192, 2	0	
conv3a	28×28×192	28×28×192	1×1×192, 1	37K	29M
conv3	28×28×192	28×28×384	3×3×192, 1	664K	521M
pool3	28×28×384	14×14×384	3×3×384, 2	0	
conv4a	14×14×384	14×14×384	1×1×384, 1	148K	29M
conv4	14×14×384	14×14×256	3×3×384, 1	885K	173M
conv5a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv5	14×14×256	14×14×256	3×3×256, 1	590K	116M
conv6a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv6	14×14×256	14×14×256	3×3×256, 1	590K	116M
pool4	14×14×256	7×7×256	3×3×256, 2	0	
concat	7×7×256	7×7×256		0	
fc1	7×7×256	1×32×128	maxout p=2	103M	103M
fc2	1×32×128	1×32×128	maxout p=2	34M	34M
fc7128	1×32×128	1×1×128		524K	0.5M
L2	1×1×128	1×1×128		0	
total				140M	1.6B

Hình 34 Cấu trúc mạng do Zeiler và Fergus đề xuất, với các lớp (layer), kích thước input (size-in) và output (size-out) có dạng $rows \times cols \times \#filters$ (dòng × cột × số lượng lọc), riêng phần nhân (kernel) là $rows \times cols, stride$ (dòng × cột, bước sải) và kích thước maxout polling là $p = 2$ [33].

kiến trúc còn lại từ mô hình Inception của GoogLeNet [27]. Ý tưởng chính của kiến trúc Inception nhằm quan sát cách các bộ phận có sẵn của đối tượng bao phủ và xấp xỉ cấu trúc thưa địa phương tối ưu hóa của mạng thị giác tích chập. Module Inception được xây dựng như trong Hình 35. Mô hình này có khoảng 6.6 – 7.5 triệu tham số và khoảng 500 triệu – 1.6 tỷ toán tử. Chi tiết mô hình xem tại Hình 36.



Hình 35 Module Inception dạng nguyên thủy (ảnh trái) và dạng giảm chiều (ảnh phải).

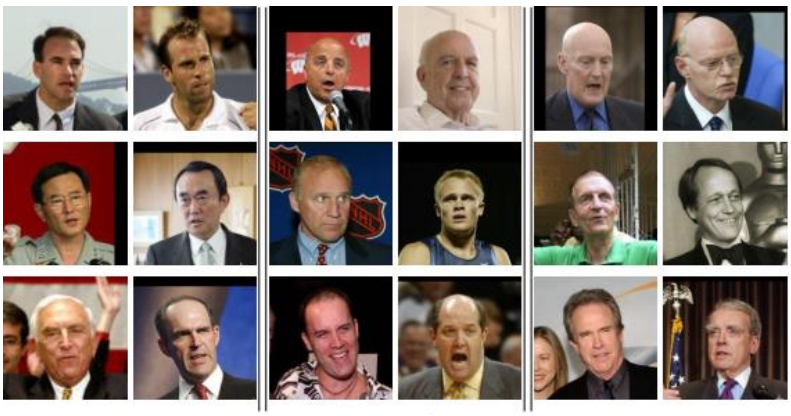
type	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj (p)	params	FLOPS
conv1 (7×7×3, 2)	112×112×64	1							9K	119M
max pool + norm	56×56×64	0						m 3×3, 2		
inception (2)	56×56×192	2		64	192				115K	360M
norm + max pool	28×28×192	0						m 3×3, 2		
inception (3a)	28×28×256	2	64	96	128	16	32	m, 32p	164K	128M
inception (3b)	28×28×320	2	64	96	128	32	64	L_2, 64p	228K	179M
inception (3c)	14×14×640	2	0	128	256,2	32	64,2	m 3×3,2	398K	108M
inception (4a)	14×14×640	2	256	96	192	32	64	L_2, 128p	545K	107M
inception (4b)	14×14×640	2	224	112	224	32	64	L_2, 128p	595K	117M
inception (4c)	14×14×640	2	192	128	256	32	64	L_2, 128p	654K	128M
inception (4d)	14×14×640	2	160	144	288	32	64	L_2, 128p	722K	142M
inception (4e)	7×7×1024	2	0	160	256,2	64	128,2	m 3×3,2	717K	56M
inception (5a)	7×7×1024	2	384	192	384	48	128	L_2, 128p	1.6M	78M
inception (5b)	7×7×1024	2	384	192	384	48	128	m, 128p	1.6M	78M
avg pool	1×1×1024	0								
fully conn	1×1×128	1							131K	0.1M
L2 normalization	1×1×128	0								
total									7.5M	1.6B

Hình 36 FaceNet sử dụng mô hình Inception tương tự với [27]. Hai điểm khác biệt chinh đó là FaceNet sử dụng L_2 pooling thay vì max pooling. Kích thước polling luôn luôn 3 × 3 và tính song song với module tích chập trong mỗi module Inception.

2.4.3. 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 của cuốn báo cáo nói về nhận dạng khuôn mặt nên tôi 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 $98.87\% \pm 0.15$ khi không canh chỉnh mặt và $99.63\% \pm 0.09$ khi có canh chỉnh mặt. Hình 37 là một số cặp ảnh nhận dạng lỗi trong bộ LFW.



Hình 37 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

khuôn mặt, khi đó độ chính xác là $95.12\% \pm 0.39$. Nếu sử dụng 1000 frame đầu tiên thì độ chính xác là 95.18%.

2.4.4. Ưu và Nhược Điểm của Thuật Toán

2.4.4.1. 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

2.4.4.2. 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. [34]

2.4.5. 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.

2.5. Nhận Dạng Khuôn Mặt Sử Dụng Thuật Toán DeepFace

2.5.1. Tóm Tắt

Nhóm tác giả [35] 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.

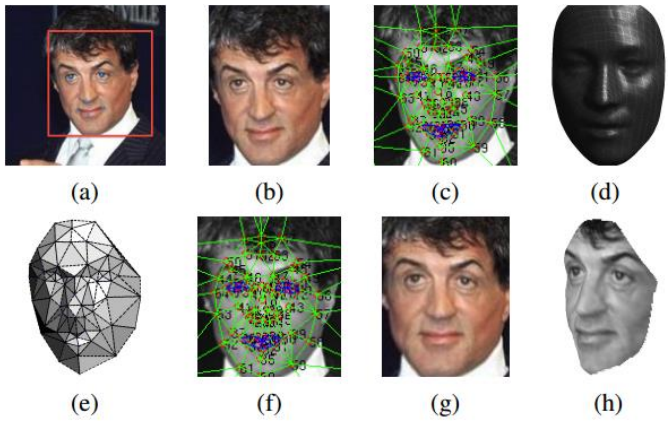
Nhóm tác giả phát triển một cấu trúc DNN hiệu quả và các tận dụng ảnh trên mạng xã hội để biểu diễn khuôn mặt sao cho có thể tổng quát hóa cho các tập dữ liệu khác. Ngoài ra, nhóm tác giả trình bày cách canh chỉnh mặt dựa trên mô hình 3D của khuôn mặt.

2.5.2. Chi Tiết Thuật Toán

2.5.2.1. Canh Chỉnh Khuôn Mặt

Gần đây có nhiều công thức có thể canh chỉnh khuôn mặt với góc chụp tùy ý, các công thức này thường phân tích mô hình 3D của khuôn mặt hoặc là tìm các dạng điểm chính tương ứng từ tập dữ liệu ngoài hoặc là công thức không giám sát nhằm tìm biến đổi tương ứng của các điểm ảnh. Mặc dù các thuật toán canh chỉnh này được sử dụng rộng rãi nhưng vẫn chưa có giải pháp phù hợp áp dụng cho mặt tự nhiên, và mô hình 3D của khuôn mặt gần đây không nhiều người sử dụng tới, nhất là trong môi trường tự nhiên. Tuy nhiên, do khuôn mặt là mô hình 3D nên nhóm tác giả quyết định đi theo con đường này.

Giống như các thuật toán trước dó, nhóm tác giả xác định các điểm chính cơ bản trên khuôn mặt để canh chỉnh, lặp nhiều lần để làm mịn kết quả output. Với mỗi lần lặp, sử dụng Support Vector Regressor (SVR) đã qua huấn luyện để dự đoán điểm chính trên khuôn mặt từ cửa sổ mô tả trên mặt dựa trên biểu đồ LBP.



Hình 38 Quy trình canh chỉnh mặt. (a) Xác định khuôn mặt với 6 điểm chính. (b) Cắt khuôn mặt. (c) 67 điểm chính từ ảnh (b) với phép đạc tam giác Delaunay tương ứng, nhóm tác giả thêm các hình tam giác vào các đường biên nhằm tránh đi tính không liên tục. (d) Hình dạng quy chiếu 3D biến dổi từ ảnh 2D trong không gian ảnh. (e) Các tam giác khớp theo camera 30-2D, tam giác càng tối càng khó thấy. (f) 67 điểm chính sinh ra từ mô hình 3D dùng để chỉnh hướng từng đoạn bao affine. (g) Cắt mặt chính diện. (h) Góc nhìn mới tạo từ mô hình 3D.

Canh chỉnh 2D. Ta bắt đầu canh chỉnh mặt bằng cách xác định 6 điểm chính trong hộp bao khuôn mặt, canh giữa bằng mắt, đỉnh mũi và miệng (Hình 38 (a)). Ba vị trí này dùng để xấp xỉ tỉ lệ, xoay và chuyển mặt thành 6 vị trí neo bằng cách gán $T_{2d}^{i} \coloneqq (s_i, R_i, t_i)$ với $x_{neo}^{j} \coloneqq s_i[R_i|t_i] * s_{nguồn}^{j}$ với các điểm $j = 1 \dots 6$ và lặp lại điều này trong ảnh bao mới cho đến khi không có sự thay đổi quan trọng nào, cuối cùng ta biến đổi tương đương 2D: $T_{2d} \coloneqq T_{2d}^{1} * \dots * T_{2d}^{k}$. Tập hợp các phép biến đổi này tạo ra mẫu ảnh 2D canh chỉnh (Hình 38 (b)).

Canh chỉnh 3D. Ta sử dụng mô hình mặt 3D và thiết lập camera affine 3D dùng để bao ảnh 2D canh chỉnh vào mặt phẳng ảnh 3D. Ta khoanh vùng thêm 67 điểm chính x_{2d} trong ảnh 2D canh chỉnh (Hình 38 (c)) sử dụng SVR thứ hai, điều này tạo ra mặt 3D đã canh chỉnh như trong Hình 38 (g). Trong mô hình chung 3D, ta chỉ cần lấy trung bình mẫu scan 3D từ bộ dữ liệu USF Human-ID [36], bộ dữ liệu này đã qua hậu xử lý và được biểu diễn theo các đỉnh đã canh chỉnh $v_i = (x_i, y_i, z_i)_{i=1}^{n}$. Ta đặt 67 điểm neo chính vào hình 3D. Sử dụng nghiệm bình phương tối thiểu tổng quát của hệ tuyến tính $x_{2d} = X_{3d}\vec{P}$ vào camera biến đổi affine P từ 3D sang 2D, hệ tuyến tính này là ma trận hiệp phương sai Σ đã biết, tức $\vec{P}$ tối thiểu hóa hàm sai số

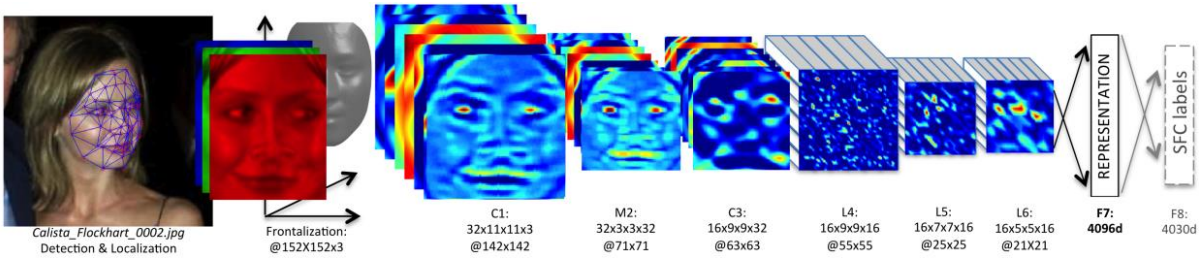
$$loss(\vec{P}) = r^T \Sigma^{-1} r \qquad (44)$$

với $r = (x_{2d} - X_{3d}\vec{P})$ là vector dư và X_{3d} là ma trận $(67 * 2) \times 8$ sinh ra bằng cách xếp dọc (2×8) ma trận $[x_{3d}^T(i), 1, \vec{0}; \vec{0}, x_{3d}^T(i), 1]$ với $\vec{0}$ là vector dòng có bốn phần tử 0, mỗi phần tử là quy chiếu cho điểm chính $x_{3d}(i)$. Ta dùng vector gồm 8 biến $\vec{P}$ để biểu diễn camera affine P có kích thước 2×4. Tối hiểu hóa hàm sai số bằng phân rã Cholesky cho Σ, biến đổi bài toán thành bài toán bình phương tối thiểu thường. Xác định các điểm chính trên biên mặt thường có nhiều nhiễu do phải ước lượng vị trí bị ảnh hưởng lớn bởi độ sâu dựa theo góc của camera, ta dùng ma trận hiệp phương sai Σ có kích thước $(67 * 2) \times (67 * 2)$ bằng cách ước lượng các hiệp phương sai từ các điểm chính lỗi.

Chính diện hóa. Do ta không mô hình phép chiếu góc nhìn đầy đủ và biến dạng không chặt, do đó P chỉ mang tính xấp xỉ. Để làm giảm lỗi những phần quan trọng khi bao mặt lần cuối, nhóm tác giả thêm phần dư r tương ứng với phần $x - y$ của mỗi điểm quy chiếu chính x_{3d}, ký hiệu là $\widetilde{x_{3d}}$. Cuối cùng, ta thu được ảnh chính diện bằng phép biến đổi affine từng phần T từ x_{2d} (ảnh nguồn) đến ảnh $\widetilde{x_{3d}}$ (ảnh mục tiêu), dùng phép đạc tam giác Delaunay từ 67 điểm chính để định hướng. Đồng thời, ta có thể thay những tam giác vô hình ứng với camera P bằng ảnh trộn với ảnh bản sao đối xứng.

2.5.2.2. Biểu Diễn

2.5.2.2.1. Cấu Trúc DNN và Huấn Luyện



Hình 39 Cấu trúc huấn luyện của DeepFace, từ ảnh vào, lấy khuôn mặt, sau đó chỉnh chính diện dựa vào mô hình 3D (Frontalization), tiếp theo là các lớp lọc Tích chập (C1) – Pooling (M2) – Tích chập (C3), sau đó là 3 lớp Liên thông Địa phương (L4 – L6) và 2 lớp Liên thông Đầy đủ (F7 – F8). Mạng có hơn 120 ngàn tham số với 95% tập trung ở Liên thông Địa phương và Liên thông Đầy đủ.

Cấu trúc huấn luyện ở Hình 39, đầu tiên ảnh vào 3D đã canh chỉnh với 3 kênh màu RGB có kích thước 152×152 điểm ảnh đưa làm lớp Tích chập (C1) với 32 bộ lọc có kích thước $11 \times 11 \times 3$ (Hình 39 ký hiệu là $32 \times 11 \times 11 \times 3@152 \times 152$). 32 ảnh đặc trưng thu được đưa vào lớp max pooling (M2) lấy giá trị lớn nhất trong mỗi khối láng giềng 3×3 ở mỗi ảnh đặc trưng với bước sải là 2. Sau đó, ta đưa vào lớp Tích chập (C3) tiếp theo gồm 16 bộ lọc có kích thước $9 \times 9 \times 16$. Mục đích thiết lập 3 lớp này nhằm trích xuất ra các đặc trưng có mức thấp như các cạnh hay kết cấu ảnh, lớp max pooling nhằm làm cho output của mạng tích chập trở nên rõ ràng hơn cho chuyển đổi địa phương. Khi áp dụng vào ảnh mặt đã canh chỉnh thì ảnh sẽ giúp cho mạng đủ chắc chắn để không bị những lỗi nhỏ ảnh hưởng. Tuy nhiên, nếu áp dụng nhiều mức pooling sẽ khiến mạng bị mất thông tin vị trí chính xác của cấu trúc chi tiết của mặt và những phần cực nhỏ trên mặt, do đó, nhóm tác giả chỉ áp dụng max pooling vào lớp Tích chập đầu tiên. 3 lớp đầu tiên có rất ít tham số, các lớp này chỉ đơn thuần mở rộng ảnh vào thành tập các đặc trưng địa phương đơn giản.

Các lớp sau đó (L4, L5 và L6) là lớp Liên thông Địa phương, giống với lớp Tích chập, các lớp này cũng áp dụng băng lọc, nhưng mỗi vị trí trong ảnh đặc trưng học một tập các lớp đặc trưng khác nhau. Do mỗi vùng trong ảnh canh chỉnh có thông kê địa phương khác nhau nên ta không đảm bảo giả thiết về tính cố định trong không gian ảnh Tích chập. Ví dụ, vùng giữa mắt và lông mày xuất hiện khác và khả năng phân biệt cao hơn vùng giữa mũi và miệng. Sử dụng các lớp địa phương không ảnh hưởng đến chi phí tính toán trích xuất đặc trưng, nhưng có ảnh hưởng đến số lượng tham số huấn luyện. Chỉ vì ta có tập dữ liệu lớn nên ta phải chịu 3 lớp Liên thông Địa phương, nguyên do vì mỗi đơn vị output của lớp Liên thông Địa phương chịu ảnh hưởng từ khối input. Ví dụ, output của L6 ảnh hưởng từ khối $74 \times 74 \times 3$ làm input và khó có mối liên hệ thống kê giữa hai khối lớn trong mặt canh chỉnh.

Cuối cùng, hai lớp F7 và F8 là lớp Liên thông Đầy đủ, trong đó mỗi đơn vị output được kết nối với tất cả input. Các lớp này có khả năng bắt được mối quan hệ đặc trưng giữa các phần xa trong khuôn mặt, ví dụ như vị trí và hình dạng mắt và vị trị, hình dạng của miệng. Sử dụng output của lớp Liên thông Đầy đủ đầu tiên (F7) trong mạng làm vector biểu diễn thô cho khuôn mặt. Xét về mặt biểu diễn, vector này khác với biểu diễn dựa trên LBP. Output của lớp Liên thông Đầy đủ cuối cùng (F8) dùng cho K-way softmax (với K là số lớp) có phân phối trên các nhãn lớp. Ta ký hiệu o_k là output thứ k của mạng khi cho trước input, xác suất gán vào lớp thứ k là output của hàm softmax

$$p_k = \frac{e^{o_k}}{\sum_h e^{o_h}} \tag{45}$$

Mục tiêu của quá trình huấn luyện nhằm tối đa xác suất của lớp chính xác (lớp của mặt) bằng cách tối thiểu hàm sai số cross-entropy cho mỗi mẫu huấn luyện. Nếu k là chỉ số đúng của lớp ảnh vào thì hàm sai số là

$$L = -\log p_k \tag{46}$$

Ta tối thiểu hàm sai số bằng cách tính độ dốc của L theo tham số và cập nhật tham số bằng Trượt Dốc Ngẫu Nhiên (Stochastic Gradient Descent – SGD). Sử dụng hàm kích hoạt ReLU $\max(0, x)$, về trung bình có 75% các phần đặc trưng ở các lớp đầu bằng 0.

Cho ảnh I, ta tính biểu diễn $G(I)$ bằng mạng truyền tiến ở trên. Có thể đánh giá bất kỳ mạng neuron truyền tiến với L lớp là phân rã của hàm g_ϕ^l. Trong trường hợp này, biểu diễn phân rã thành

$$G(I) = g_\phi^{F_7}\left(g_\phi^{L_6}\left(\dots g_\phi^{C_1}\left(T(I, \theta_T)\right)\dots\right)\right) \tag{47}$$

với tham số $\phi = \{C_1, \dots, F_7\}$ và $\theta_T = \{x_{2d}, \vec{P}, \vec{r}\}$.

2.5.2.2.2. Chuẩn Hóa

Bước cuối cùng, ta chuẩn hóa đặc trưng về miền quá trị 0 đến 1 nhằm giảm sự thay đổi độ nhạy sáng: Chia mỗi phần tử trong vector đặc trưng cho giá trị lớn nhất trong suốt quá trình huấn luyện, thực hiện điều này bằng chuẩn hóa L_2

$$f(I) \coloneqq \frac{\bar{G}(I)}{\|\bar{G}(I)\|_2} \tag{48}$$

trong đó

$$\bar{G}(I)_i = \frac{G(I)_i}{\max(G_i, \epsilon)} \tag{49}$$

giá trị ϵ ở đây nhằm tránh trường hợp chia cho số rất nhỏ (nhóm tác giả chọn $\epsilon = 0.05$).

2.5.2.3. Metric dùng để Kiểm Tra

Nhóm tác giả học metric không giám sát nhằm xác định mức độ giống nhau giữa 2 khuôn mặt bằng cách lấy tích trong của 2 vector đặc trưng chuẩn hóa. Nhóm thực nghiệm phép đo này với metric có giám sát là khoảng cách χ^2 và mạng Siamese.

2.5.2.3.1. Khoảng Cách χ^2 có Trọng Số

Vector đặc trưng chuẩn hóa DeepFace có nhiều nét tương dồng với đặc trưng dựa trên biểu đồ như LBP như chứa các giá trị không âm, giá trị thưa và nằm trong đoạn $[0,1]$. Do đó, nhóm tác giả sử dụng khoảng cách χ^2 có trọng số

$$\chi^2(f_1, f_2) = \sum_i \frac{w_i(f_1[i] - f_2[i])^2}{f_1[i] + f_2[i]} \quad (50)$$

với f_1 và f_2 là vector đặc trưng của DeepFace, sử dụng SVM tuyến tính để học trọng số, áp dụng vào vector với các phần tử $(f_1[i] - f_2[i])^2/(f_1[i] + f_2[i])$.

2.5.2.3.2. Mạng Siamese

Nhóm tác giả đồng thời kiểm tra trên metric đầu-đến-cuối tên là mạng Siamese [37]: Sau khi học, lặp lại mạng nhận dạng khuôn mặt 2 lần (không dùng lớp trên cùng, mỗi lần cho 1 ảnh input) và đặc trưng dùng để dự đoán trực tiếp 2 input có thuộc về cùng một người hay không. Ta thực hiện điều này bằng cách:

a) Lấy sai số tuyệt đối của 2 đặc trưng,
b) Lớp Liên thông Đầy đủ trên cùng ánh xạ vào đơn vị logistic đơn (giống/không giống).

Để ngăn chặn hiện tượng overfitting khi nhận dạng, ta chỉ huấn luyện 2 lớp trên cùng. Khoảng các của mạng Siamese là

$$d(f_1, f_2) = \sum_i \alpha_i |f_1[i] - f_2[i]| \quad (51)$$

trong đó α_1 là tham số huấn luyện. Tham số trng mạng Siamese được huấn luyện bởi hàm sai số cross entropy tiêu chuẩn và lan truyền ngược lỗi.

2.5.3. Thực Nghiệm

Nhóm tác giả thực nghiệm sử dụng một CPU Intel 2.2GHz với các bộ dữ liệu:

- Bộ dữ liệu LFW: Gồm 13323 ảnh từ 5749 người nổi tiếng.
- Bộ dữ liệu YouTubeFace: gồm 3425 video trên Youtube của 1595 đối tượng.

Thực nghiệm trên LFW, sử dụng metric không giám sát nhằm so sánh trực tiếp tích trong của cặp vector đặc trưng, độ chính xác trung bình thu được là 95.92%. Tiếp theo, nhóm tác giả học nhân SVM và dùng khoảng cách χ^2 thì độ chính xác là 97.00%.

Thực nghiệm trên YouTubeFace bằng cách biểu diễn DeepFace trực tiếp trên mỗi cặp video, 50 cặp frame, mỗi phần từ một video và dán nhãn giống hoặc không giống, sau đó học trọng số mô hình χ^2. Cho cặp kiểm tra, nhóm tác giả lấy mẫu ngẫu nhiên 100 cặp frame, mỗi phần từ 1 video và dùng giá trị trung bình của trọng số học được để đánh giá mức tương đồng, kết quả độ chính xác trung bình là 91.4%.

2.5.4. Ưu và Nhược Điểm của Thuật Toán

2.5.4.1. Ưu Điểm

Đến thời điểm hiện tại, DeepFace là một trong những thuật toán nhận dạng khuôn mặt có độ chính xác thuộc dạng "top performing".

2.5.4.2. Nhược Điểm

DeepFace huấn luyện với bộ dữ liệu riêng, bao gồm hàng triệu ảnh truyền thông, xã hội có kích thước lớn hơn các bộ dữ liệu hiện hữu trong nghiên cứu học thuật. [38]

2.5.5. 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.

3. BỘ DỮ LIỆU SỬ DỤNG CHO ĐỀ TÀI

3.1. Bộ Dữ Liệu PIE

Bộ dữ liệu PIE do Viện Robotics, trường Đại học Carnegie Mellon thiết lập vào năm 2003, gồm 41368 hình của 68 người, ảnh có kích thước 640 × 486 gồm ảnh chân dung, ảnh sáng, ảnh cảm xúc (xem Hình 40) [39].

Hình 40 Ví dụ ảnh trong bộ dữ liệu PIE gồm: Ảnh chân dung, ảnh sáng, ảnh cảm xúc.

3.2. Bộ Dữ Liệu UMIST

Bộ dữ liệu UMIST từ trường Đại học Sheffield gồm 564 ảnh từ 20 đối tượng, mỗi đối tượng gồm ảnh có góc chụp mặt bên phải xoay sang chính diện, ảnh là ảnh xám có kích thước 220 × 220 điểm ảnh (xem Hình 41) [40].

Hình 41 Ảnh trong bộ dữ liệu UMIST chụp từ góc mặt phải sang mặt chính diện.

3.3. Bộ Dữ Liệu CVL

Bộ dữ liệu CVL từ Phòng Thí nghiệm Thị giác Máy tính, trường Đại học Ljubljana, Slovenia gồm 114 người, mỗi người có 5 ảnh chụp từ góc mặt bên phải sang mặt bên trái và 2 ảnh cảm xúc khuôn mặt với độ phân giải 640×480 điểm ảnh (xem Hình 42).

Hình 42 Ảnh trong bộ dữ liệu CVL. Ảnh trên: 5 ảnh chụp góc mặt từ phải sang trái. Ảnh dưới: Ảnh cảm xúc khuôn mặt

4. HƯỚNG PHÁT TRIỂN TIẾP THEO

Sau khi tìm hiểu 5 bài báo chính, ta tạm có 2 hướng phát triển

- **Hướng phát triển 1.** Sử dụng ý tưởng ở 2.1, trong đó khuôn mặt sẽ chia theo superpixel, xác định điểm chính dựa vào thuật toán ở 2.3, từ đó sẽ cắt ra thành các phần mặt. Ở mỗi phần mặt, dùng đặc trưng sao cho có thể lấy được những chi tiết nhỏ trong phần mặt đó (ví dụ như LBP), thiết lập biểu đồ, ta thu được vector đặc trưng ở mỗi phần mặt. Mặt kiểm tra ta thực hiện tương tự, với mỗi phần mặt trong mặt kiểm tra, ta tìm phần mặt tương ứng trong bộ dữ liệu, đối tượng nào có số lượng phần mặt xuất hiện nhiều nhất, ta sẽ suy ra ảnh kiểm tra là ảnh của đối tượng đó (xem Hình 27). Ý tưởng phần kiểm tra tương đồng với ý tưởng của Bag of Word (xem 2.1.2.1).
- **Hướng phát triển 2.** Sử dụng DNN của FaceNet và DeepFace để nhận dạng khuôn mặt, tuy nhiên ta sẽ thay đổi cấu trúc lọc của 2 thuật toán này, có thể thêm một số bộ lọc khác như Contourlet để làm rõ các đường biên, LBP để lấy các chi tiết nhỏ.

TÀI LIỆU THAM KHẢO

[1] Z. Li, J.-i. Imai and M. Kaneko, "Robust face recognition using block-based bag of words.," *Pattern Recognition (ICPR), 2010 20th International Conference on. IEEE,* pp. 1285-1288, 2010.

[2] C.-F. Tsai, "Bag-of-words representation in image annotation: A review.," *ISRN Artificial Intelligence 2012,* 2012.

[3] L. Fei-Fei, "Stanford University, Computer Vision Lab," 6 9 2012. [Online]. Available: http://vision.stanford.edu/teaching/cs231a_autumn1112/lecture/lecture14_intro_objrecog_bow_cs231a.pdf.

[4] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories.," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05),* vol. Vol. 2, pp. 524-531, 2005.

[5] A. Martinez and R. Benavente, "The AR Face Database," *CVC Technical Report #24,* June 1998.

[6] S. Liao, A. K. Jain and S. Z. Li, "Partial face recognition: Alignment-free approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence 35.5 ,* pp. 1193-1205, 2013.

[7] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE transactions on pattern analysis and machine intelligence,* vol. 34.11, pp. 2274-2282, 2012.

[8] J. Y. Wright, G. A. Y., S. S. S. A. and Y. Ma, "Robust face recognition via sparse representation," *IEEE transactions on pattern analysis and machine intelligence,* vol. 31(2), pp. 210-227, 2009.

[9] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision,* vol. 60.2, pp. 91-110, 2004.

[10] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence,* vol. 6, pp. 679-698, 1986.

[11] K. Mikolajczyk, A. Zisserman and C. Schmid, "Shape recognition with edge-based features," *British Machine Vision Conference (BMVC'03),* vol. Vol. 2, pp. 779-788, 2003.

[12] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *International journal of computer vision,* vol. 60(1), pp. 63-86, 2004.

[13] T. Lindeberg, "Feature detection with automatic scale selection," *International journal of computer vision,* vol. 30.2, pp. 79-116, 1998.

[14] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *IEEE transactions on image processing,* vol. 19.6, pp. 1635-1650, 2010.

[15] D. L. Donoho and Y. Tsaig, "Fast solution of-norm minimization problems when the solution may be sparse," *IEEE Transactions on Information Theory,* vol. 54.11, pp. 4789-4812, 2008.

[16] C. Ding and D. Tao, "A comprehensive survey on pose-invariant face recognition," *ACM Transactions on Intelligent Systems and Technology (TIST),* vol. 7.3, p. 37, 2016.

[17] R. Weng, J. Lu, J. Hu, G. Yang and Y. P. Tan, "Robust feature set matching for partial face recognition," *Proceedings of the IEEE International Conference on Computer Vision ,* pp. 601-608, 2013.

[18] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman and N. Kumar, "Localizing parts of faces using a consensus of exemplars," *IEEE transactions on pattern analysis and machine intelligence,* vol. 35(12), pp. 2930-2940, 2013.

[19] Reinders, M. JT, R. W. C. Koch and J. J. Gerbrands, "Locating facial features in image sequences using neural networks," *Automatic Face and Gesture Recognition, Proceedings of the Second International Conference on. IEEE,* pp. 230-235, 1996.

[20] P. Campadelli, G. Lipori and R. Lanzarotti, Automatic facial feature extraction for face recognition, INTECH Open Access Publisher, 2007.

[21] X. Cao, Y. Wei, F. Wen and J. Sun, "Face alignment by explicit shape regression," *International Journal of Computer Vision ,* vol. 107.2, pp. 177-190, 2014.

[22] M. Dantone, J. Gall, G. Fanelli and L. Van Gool, "Real-time facial feature detection using conditional regression forests," *Computer Vision and Pattern Recognition (CVPR),* pp. 2578-2585, June, 2012.

[23] F. Schroff, D. Kalenichenko and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 815-823, 2015.

[24] B. Amos, "OpenFace," [Online]. Available: https://cmusatyalab.github.io/openface/.

[25] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *Journal of Machine Learning Research,* vol. 10, pp. 207-244, 2009.

[26] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *European Conference on Computer Vision, Springer International Publishing,* pp. 818-833, 2014.

[27] C. Szegedy and e. al, "Going deeper with convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 1-9, 2015.

[28] D. R. Wilson and T. R. Martinez, "The general inefficiency of batch training for gradient descent learning," *Neural Networks,* vol. 16.10, pp. 1429-1451, 2003.

[29] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation,* vol. 1 (4), pp. 541-551, 1989.

[30] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning representations by back-propagating errors," *Cognitive modeling,* vol. 5(3), 1988.

[31] J. Duchi, E. Hazan and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research,* vol. 12, pp. 2121-2159, 2011.

[32] A. Geitgey, "medium.com," A Medium Corporation, [Online]. Available: https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78#.iyo9udyws. [Accessed 24 July 2016].

[33] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. C. Courville and Y. & Bengio, "Maxout networks," *ICML* , vol. 3, pp. 1319-1327, 2013.

[34] O. M. Parkhi, A. Vedaldi and A. Zisserman, "Deep face recognition," *British Machine Vision Conference,* vol. 1, no. 3, 2015.

[35] Y. Taigman, M. Yang, Ranzato, M. A. and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 1701-1708, 2014.

[36] S. Sarkar, "USF Human ID 3-D Database," [Online]. Available: http://www.cse.usf.edu/~sarkar/SudeepSarkar/3D_Face_Data.html.

[37] S. Chopra, R. Hadsell and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05),* vol. 1, pp. 539-546, 2005.

[38] B. Amos, B. Ludwiczuk and M. Satyanarayanan, "OpenFace: A general-purpose face recognition library with mobile applications," CMU School of Computer Science, 2016.

[39] T. Sim, S. Baker and M. Bsat, "The CMU pose, illumination, and expression (PIE) database.," *Automatic Face and Gesture Recognition, 2002. Proceedings,* vol. Fifth IEEE International Conference, pp. 46-51, 2002.

[40] I. E. Laboratory, the University of Sheffield, [Online]. Available: https://www.sheffield.ac.uk/eee/research/iel/research/face.