TRƯỜNG ĐẠI HỌC SÀI GÒN
TẠP CHÍ KHOA HỌC
ĐẠI HỌC SÀI GÒN
ISSN 1859 - 3208
Số 85 (02/2023)

SAIGON UNIVERSITY
SCIENTIFIC JOURNAL
OF SAIGON UNIVERSITY
No. 85 (02/2023)

Email: tcdhsg@sgu.edu.vn ; Website: http://sj.sgu.edu.vn/

XÂY DỰNG HỆ THỐNG NHẬN DẠNG MẶT NGƯỜI THEO THỜI GIAN THỰC

Building a real time faces recognition system

Đoàn Chánh Tín[(1)], Trần Thị Hoàng Oanh[(2)], Nguyễn Duy Phương[(3)*]

[(1)]Trường Cao đẳng Công nghệ Thủ Đức
[(2)]Trường Đại học Kinh tế - Kỹ thuật Bình Dương
[(3)]Trường Đại học Sài Gòn

TÓM TẮT

Hiện nay, hầu hết các công ty, xí nghiệp đều có hệ thống camera giám sát hoạt động 24/7, đồng nghĩa với việc phải có nhân viên túc trực để giám sát camera, điều này dẫn đến hao phí tài nguyên con người, và đôi khi do mệt mỏi hay sơ ý người nhân viên không quan sát camera thì có thể bỏ sót tội phạm. Điển hình là hàng loạt các vụ đột nhập vào hộ gia đình, cửa hàng, văn phòng… tuy được camera ghi nhận lại hình ảnh nhưng không có người phát hiện kịp thời, nên những tên tội phạm vẫn có thể ung dung hoạt động và trở ra an toàn. Đây chính là mặt hạn chế của hệ thống giám sát hiện nay.

Xuất phát từ thực tiễn đó, bài báo này nghiên cứu phát triển một hệ thống giám sát cảnh báo hoạt động hiệu quả hơn, chính xác hơn mà không cần có sự can thiệp của con người. Hệ thống hoạt động trên thời gian thực, giám sát liên tục và đồng thời phát hiện, nhận dạng nhiều đối tượng, phát hiện người lạ đi vào cơ quan.

Bài báo áp dụng những thành tựu khoa học kỹ thuật sẵn có như camera giám sát an ninh, kết hợp với kỹ thuật xử lý ảnh bằng thư viện thị giác máy tính mã nguồn mở OpenCV để xử lý hình ảnh thu nhận. Bài báo là sự kết hợp của những kiến thức cơ bản về xử lý ảnh số, lập trình Visual C++ với thư viện OpenCV.

Từ khóa: *nhận dạng mặt người, OpenCV, PCA, xử lý ảnh*

ABSTRACT

Today, almost all companies, enterprises have surveillance security camera system that monitoring their business 24/7, so it requires human intervention. This is leading to the wastage of human resources, and sometime if the person who was monitoring the camera was tire or careless then the crime might be overlook. Typically, there were many crimes that the thieves break into household and office that have surveillance security camera system, but they were able to flee freely from the crime scene because no one was there to reports the crime. The surveillance camera alerting side is very limited in the security monitoring system today because it cannot be automatic and it requires human intervation. This is the limited side of the security monitoring system today.

Stemming for this limit of the surveillance security camera system today, my dissertation research is to developed a monitoring and alerting security camera system that is more efficient and it doesn't need human intervention. The security camera system that I am developed can operate the system on real-time, it can continuously and simultaneously monitoring and detecting object, and the recognition of the object is based on multiple databases available.

This thesis applies the scientific and technical achievements of security surveillance camera that are available today; combined with image processing techniques in computer vision library open source OpenCV to process the images that obtained. My thesis is a combination of the basic knowledge of digital image processing, Visual C + + programming with OpenCV libraries.

Keywords: *faces recognition, OpenCV, PCA, image processing*

*Tác giả liên hệ: phuong.nd@sgu.edu.vn

1. Đặt vấn đề

Trước sự phát triển mạnh mẽ của khoa học kỹ thuật, ngành điều khiển học và tự động hóa đã có những bước tiến lớn. Điều khiển học và tự động hóa ngày càng đóng vai trò quan trọng trong việc tăng năng xuất lao động, giảm giá thành, tăng sự ổn định chất lượng của sản phẩm. Thêm vào đó, môi trường làm việc được cải thiện đặc biệt trong một số ngành nghề có tính chất độc hại nguy hiểm cho tính mạng con người: dò bom mìn, thao tác trong các nhà máy điện hạt nhân hoặc các khu vực có độ phóng xạ cao, thám hiểm mặt trăng sao hỏa, thám hiểm đáy biển sâu. Trong xã hội công nghiệp ngày nay, việc ứng dụng các robot tự hành, các cánh tay máy, có khả năng hoạt động độc lập ngày càng phổ biến. Để có khả năng thông minh đó, chúng phải có khả năng nhìn, nhận dạng các tình huống, xử lý tình huống như con người. Đáp ứng nhu cầu đó, xử lý ảnh đã tham gia vào điều khiển học và tự động hóa. Dữ liệu về hình ảnh có lượng thông tin lớn và chính xác hơn các loại dữ liệu từ các cảm biến khác. Điều này giúp cho việc giải quyết các vấn đề trong lĩnh vực điều khiển tự động dễ dàng hơn. Một trong những bài toán về xử lý ảnh phổ biến hiện nay là "phát hiện và nhận dạng khuôn mặt người" và đã được ứng dụng rộng rãi vào cuộc sống hiện nay như:

- Hệ thống tương tác giữa người và máy: giúp những người bị tật hoặc khuyết điểm có thể trao đổi. Những người dùng ngôn ngữ tay có thể giao tiếp với người bình thường. Những người bị bại liệt thông qua một số ký hiệu nháy mắt có thể biểu lộ những gì họ muốn. Đó là các bài toán điệu bộ của bàn tay (hand gesture), điệu bộ của khuôn mặt.
- Nhận dạng người A có phải là tội phạm truy nã hay không, giúp cơ quan an ninh quản lý tốt con người. Công việc nhận dạng có thể ở trong môi trường bình thường cũng như trong bóng tối.
- Thẻ căn cước, chứng minh nhân dân (Face identification).
- An ninh sân bay, xuất nhập cảnh (hiện nay cơ quan an ninh xuất nhập cảnh Mỹ đã áp dụng). Dùng để xác thực người xuất nhập cảnh và kiểm tra có phải là nhân vật khủng bố hay không.
- Điều khiển vào ra: văn phòng, công ty, trụ sở, máy tính… kết hợp thêm vân tay và móng mắt. Cho phép nhân viên được vào ra nơi cần thiết, hay mỗi người sẽ đăng nhập máy tính cá nhân của mình mà không cần nhớ tên đăng nhập cũng như mật khẩu mà chỉ cần xác định thông qua khuôn mặt.
- Tìm kiếm dữ liệu liên quan đến con người thông qua khuôn mặt người trên nhiều hệ cơ sở dữ liệu lưu trữ thật lớn như internet, các hãng truyền hình.

Để phát hiện, nhận diện và cảnh báo, đảm bảo tính chính xác thì hệ thống cần phải có tốc độ xử lý nhanh, hoạt động như một hệ thống thời gian thực. Vì vậy tốc độ nhanh và tính chính xác là 2 yếu tố quan trọng nhất trong việc xây dựng và thực hiện hệ thống.

Vì vậy, bài báo trình bày về phương pháp xây dựng một phần mềm có thể phát hiện và nhận dạng khuôn mặt người trong thời gian thực dựa trên thuật toán phân tích thành phần chính PCA để nhận dạng khuôn mặt người, để từ đó phân biệt người làm việc tại đơn vị và người lạ bên ngoài vào.

Các kết quả nghiên cứu trong nước:

- "Nhận dạng khuôn mặt sử dụng phương pháp biến đổi Eigenfaces và mạng Nơron" của Trương Công Lợi. Trong đề tài tác giả đã sử dụng thuật toán PCA để nhận dạng khuôn mặt, từ đó xây dựng chương trình tìm kiếm một bức ảnh có khuôn mặt một người trong tập ảnh cơ sở giống với

khuôn mặt của người trong bức ảnh cần kiểm tra bằng ngôn ngữ Matlab [1].

- "Xác thực sinh trắc học khuôn mặt trên thiết bị di động" của Mạch Thị Kim Hạnh. Tác giả thực hiện nghiên cứu các phương pháp rút trích đặc trưng khuôn mặt Eigenfaces, Fisherfaces, Local Binary Pattern và các hệ thống xác thực, bảo mật trên di động để từ đó xây dựng một giao diện ứng dụng trên điện thoại di động cho phép người dùng xác thực bằng khuôn mặt [2].

Các kết quả nghiên cứu ngoài nước:

- "Faces Regconition Using Eigenfaces" của Matthew A.Turk và Alex P. Pentland. Trong bài báo này hai tác giả đã xây dựng thuật toán cho phép phát hiện và nhận dạng mặt người theo thời gian thực dựa trên kỹ thuật rút trích đặc trưng các thành phần chính của khuôn mặt [3].

- "Data analysis using principal component analysis" của Shruti Sehgal, Harpreet Singh, Mohit Agarwal, V. Bhasker, Shantanu. Các tác giả đã đề xuất một thuật toán dùng để giảm kích thước của tập dữ liệu gốc ban đầu từ dữ liệu 11 chiều sang 2 chiều. Mục tiêu của thuật toán là giới hạn thông tin tối đa chỉ trong hai cột được gọi là thành phần chính và bỏ qua phần còn lại của các cột mang số lượng thông tin không đáng kể. Để giảm số chiều của dữ liệu, các tác giả đã sử dụng thuật toán phân tích thành phần chính PCA và cho kết quả tốt. Qua đó, các tác giả có được một cái nhìn rõ ràng về mối tương quan giữa các biến khác nhau khi chúng được thể hiện lên biểu đồ 2D trong phần mềm MATLAB [4].

2. Phương pháp nghiên cứu

2.1. Các phương pháp nhận dạng mặt người

Phương pháp nhận dạng mặt người hiện nay có 2 loại:

- Nhận dạng dựa trên các đặc trưng của các phần tử trên khuôn mặt (Feature based face recognition).
- Nhận dạng dựa trên xét tổng thể khuôn mặt (Appearance based face recognition).

Ngoài ra còn có một số loại nhận dạng sử dụng mô hình về khuôn mặt, một số phương pháp được dùng cho loại này:

- Nhận dạng 2D: Elastic Bunch Graph, Active Appearance Model [5].
- Nhận dạng 3D: 3D Morphable Model [5].

a. Nhận dạng dựa trên các đặc trưng khuôn mặt

Đây là phương pháp nhận dạng khuôn mặt dựa trên việc xác định các đặc trưng hình học của các chi tiết trên một khuôn mặt (như vị trí, diện tích, hình dạng của mắt, mũi, miệng…) và mối quan hệ giữa chúng (như khoảng cách của hai mắt, khoảng cách của hai lông mày…).

Ưu điểm của phương pháp này là nó gần với cách mà con người sử dụng để nhận biết khuôn mặt. Hơn nữa với việc xác định đặc tính và các mối quan hệ, phương pháp này có thể cho kết quả tốt trong các trường hợp ảnh có nhiều nhiễu như bị nghiêng, bị xoay hoặc ánh sáng thay đổi.

Nhược điểm của phương pháp này là cài đặt thuật toán phức tạp do việc xác định mối quan hệ giữa các đặc tính đòi hỏi các thuật toán phức tạp. Mặt khác, với các ảnh kích thước bé thì các đặc tính sẽ khó phân biệt do đó khó tính toán được các đặc trưng trên khuôn mặt.

b. Nhận dạng dựa trên xét toàn diện khuôn mặt

Nhận dạng dựa trên toàn diện khuôn mặt, có nghĩa sẽ không đi xét đến từng thành phần đặc trưng trên khuôn mặt nữa, mà sẽ xem khuôn mặt là một không gian cụ thể và tìm những đặc trưng, những đặc điểm chính

trên không gian đó.

Nội dung chính của hướng tiếp cận này là xem mỗi ảnh có kích thước RxC là một vector trong không gian RxC chiều. Ta sẽ xây dựng một không gian mới có chiều nhỏ hơn sao cho khi biểu diễn trong không gian đó các đặc điểm chính của một khuôn mặt không bị mất đi. Trong không gian đó, các ảnh của cùng một người sẽ được tập trung lại thành một nhóm gần nhau và cách xa các nhóm khác.

Hai phương pháp thường được sử dụng là trong hướng tiếp cận này:

- PCA (Principle Components Analysis) [5]: Phân tích thành phần chính
- LDA (Linear Discriminant Analysis) [5]: Phân tích sự phân biệt tuyến tính

Ưu điểm của phương pháp phân tích thành phần chính PCA

- Giảm số chiều không gian dữ liệu xuống còn 2 chiều, giúp dễ dàng quan sát và hiểu rõ dữ liệu hơn.
- Khi tập dữ liệu có quá nhiều đặc trưng, hiệu suất của thuật toán nhận dạng sẽ giảm sút nghiêm trọng. PCA là một phương pháp rất phổ biến để tăng tốc độ học máy bằng cách loại bỏ các biến không đóng góp vào bất kỳ quá trình ra quyết định nào.
- Giúp giảm sức mạnh tính toán phần cứng do chỉ giữ lại các thành phần chính

Do đó, trong đề tài này tác giả sử dụng phương pháp phân tích thành phần chính PCA để nhận dạng khuôn mặt người.

2.2. Thuật toán phân tích thành phần chính PCA

PCA (Principle Components Analysis) là một thuật toán được sử dụng để tạo ra một ảnh mới từ ảnh ban đầu. Ảnh mới này có kích thước nhỏ hơn rất nhiều so với ảnh đầu vào và vẫn mang những đặc trưng cơ bản nhất của ảnh cần nhận dạng. PCA không cần quan tâm đến việc tìm ra các đặc điểm cụ thể của thực thể cần nhận dạng và mối quan hệ giữa các đặc điểm đó. Tất cả các chi tiết đó đều được thể hiện ở ảnh mới được tạo ra từ PCA.

Mục tiêu của phương pháp PCA là "giảm số chiều" của 1 tập vector sao cho vẫn đảm bảo được "tối đa thông tin quan trọng nhất". Tức Feature extraction [6] (giữ k thuộc tính "mới") chứ không phải Feature selection [6] (giữ lại k thuộc tính nguyên gốc ban đầu).

Tuy nhiên, khi loại bỏ một số thành phần của *x* để thu được *y* thì sẽ gây ra sai số (lỗi). Phương pháp PCA sẽ cố gắng tìm phép biến đổi tuyến tính *T* thỏa: $y = Tx$ sao cho trung bình bình phương lỗi (MSE) là bé nhất.

Gọi M là vector trung bình của tất cả các vector *x* trong tập mẫu. Gọi ma trận hiệp phương sai của các phần tử *x* trong tập mẫu là C. C được tính theo công thức sau:

$$C = \frac{1}{M-1}\sum_{k=1}^{M}(x_k - M)(x_k - M)^T \quad (1)$$

Nếu T là ma trận *m* hàng, mỗi hàng là 1 vector riêng của C, đồng thời *m* vector riêng này phải ứng với *m* trị riêng lớn nhất. Khi đó T chính là phép biến đổi thỏa MSE nhỏ nhất.

Như vậy, phương pháp PCA quy về việc đi tìm trị riêng (eigenvalues) [6] và vector riêng (eigenvectors) [6] của ma trận hiệp phương sai C của tập mẫu *X*. Sau đó, ta chỉ giữ lại K vector riêng ứng với K trị riêng lớn nhất để làm cơ sở cho không gian mới này.

2.3. Minh họa các bước tính toán Eigenfaces

Bước 1: Sử dụng các ảnh khuôn mặt $M_1, M_2, \ldots M_n$ (tập các khuôn mặt huấn luyện) với khuôn mặt phải *chính diện* và tất

cả ảnh phải *cùng kích thước*. Trong bài báo, cơ sở dữ liệu được bao gồm ảnh của tác giả chụp trực tiếp từ camera và ảnh của nghệ sĩ Hoài Linh lấy từ internet, các mặt nghiên đã được xoay về mặt chính diện. Kích thước mỗi ảnh là 200x200 pixel.



Hình 2.1. *Tập mẫu khuôn mặt*

<u>Bước 2:</u> Biểu diễn mọi ảnh M_i thành vector Γ_i

- Trong bài báo tác giả sử dụng ảnh có kích thước 200x200 để huấn luyện nhưng để minh họa và trình bày thuật toán trong bài báo tác giả đã lấy ví dụ ảnh có kích thước 3x3 và có 4 ảnh. Do kích thước ma trận 200x200 là quá lớn để thể hiện trong bài báo.

M_1 =	M_2 =	M_3 =	M_4 =
120 225 90	158 217 29	190 48 168	236 247 120
251 50 158	254 12 255	69 0 255	254 0 254
132 0 225	19 230 8	240 98 176	210 225 250

Γ_1 =	Γ_2 =	Γ_3 =	Γ_4 =
120	158	190	236
225	217	48	247
90	29	168	120
251	254	69	254
50	12	0	0
158	255	255	254
132	19	240	210
0	230	98	225
225	8	176	250

Bước 3: Tính vector khuôn mặt trung bình Ψ theo công thức:

$$\Psi = \frac{1}{M}\sum_{i=1}^{M}\Gamma_i \qquad (2)$$

Cụ thể ta có:

Ψ =

176.0000
184.2500
101.7500
207.0000
15.5000
230.5000
150.2500
138.2500
164.7500

Bước 4: Trừ vector khuôn mặt trung bình:

$$\Phi_i = \Gamma_i - \Psi \qquad (3)$$

Cụ thể ta có:

Φ_1 =	Φ_2 =	Φ_3 =	Φ_4 =
-56.0000	-18.0000	14.0000	60.0000
40.7500	32.7500	-136.2500	62.7500
-11.7500	-72.7500	66.2500	18.2500
44.0000	47.0000	-138.0000	47.0000
34.5000	-3.5000	-15.5000	-15.5000
-72.5000	24.5000	24.5000	23.5000
-18.2500	-131.2500	89.7500	59.7500
-138.2500	91.7500	-40.2500	86.7500
60.2500	-156.7500	11.2500	85.2500

Bước 5: Tính ma trận hiệp phương sai (covariance) C:

$$C = \frac{1}{M}\sum_{n=1}^{M}\Phi_n\Phi_n^T = \frac{1}{M}.A.A^T \qquad (4)$$

C sẽ có kích thước N^2xN^2

Trong đó: $A = [\Phi_1\Phi_2 \ldots \Phi_n]$

A sẽ có kích thước là N^2xM

Cụ thể ta có: Ma trận A kích thước N^2xM *(9x4)* tạo bởi ghép các Φ_i làm các cột

A =

-56.0000	-18.0000	14.0000	60.0000
40.7500	32.7500	-136.2500	62.7500
-11.7500	-72.7500	66.2500	18.2500
44.0000	47.0000	-138.0000	47.0000
34.5000	-3.5000	-15.5000	-15.5000
-72.5000	24.5000	24.5000	23.5000
-18.2500	-131.2500	89.7500	59.7500
-138.2500	91.7500	-40.2500	86.7500
60.2500	-156.7500	11.2500	85.2500

Ma trận chuyển vị của A:

A^T =

-56.00	40.75	-11.75	44.00	34.50	-72.50	-18.25	-138.25	60.25
-18.00	32.75	-72.75	47.00	-3.50	24.50	-131.25	91.75	-156.75
14.00	-136.25	66.25	-138.00	-15.50	24.50	89.75	-40.25	11.2500
60.00	62.75	18.25	47.00	-15.50	23.50	59.75	86.75	85.25

Từ đó ta dễ dàng tính được ma trận hiệp phương sai C, kết quả như sau:

C =

1814.00	-253.50	997.50	-605.50	-754.00	1343.00	2056.50	2683.00	1180.00
-253.50	6308.69	-2685.69	6271.00	607.63	-1003.88	-3380.31	2074.69	284.56
997.50	-2685.69	2538.19	-3055.25	-365.13	280.38	4199.81	-1533.44	3249.19
-605.50	6271.00	-3055.25	6349.50	691.00	-1078.75	-4137.25	1965.25	-565.50
-754.00	607.63	-365.13	691.00	420.75	-832.75	-621.88	-1452.88	282.88
1343.00	-1003.88	280.38	1078.75	-832.75	1752.25	427.63	3330.88	-1482.38
2056.50	-3380.31	4199.81	-4137.25	-621.88	427.63	7296.19	-1987.06	6394.31
2683.00	2074.69	-1533.44	1965.25	-1452.88	3330.88	-1987.06	9169.19	-3942.19
1180.00	284.56	3249.19	-565.50	282.88	-1482.38	6394.31	-3942.19	8898.69

Ma trận hiệp phương sai C có kích thước N^2xN^2 (9x9). (Nếu dùng ảnh từ tập huấn luyện 200x200 thì ma trận hiệp phương sai C sẽ có kích thước 200^2x200^2).

<u>Bước 6:</u> Tính các Eigenvector u_i (*"vector riêng"*) của ma trận vuông $A.A^T$ (C có kích thước N^2xN^2).

Do Ma trận C này có kích thước quá lớn nên việc tính các vector riêng sẽ tốn rất nhiều thời gian.

Ma trận $A.A^T$ có thể có đến N^2 trị riêng (mỗi trị riêng sẽ ứng với vô số vector riêng

– nó được gọi là *"không gian riêng ứng với trị riêng nào đó [4]"*).

Do đó ta sẽ chỉ tìm các vector riêng ứng với các trị riêng lớn nhất của ma trận $A.A^T$

- **Bước 6.1:** Xét ma trận $A^T.A$ (ma trận này chỉ có kích thước là *MxM*) có thể có M trị riêng. *M* trị riêng của ma trận $A^T.A$ (kèm với các vector riêng tương ứng), sẽ ứng với *M* trị riêng lớn nhất của $A.A^T$

Cụ thể ta có:

$A^T.A$ =

1.0e+04 *

3.6393	-1.6365	-1.0893	-0.9135
-1.6365	5.9726	-3.2601	-1.0759
-1.0893	-3.2601	5.2835	-0.9341
-0.9135	-1.0759	-0.9341	2.9235

Ma trận $A^T.A$ có kích thước *4x4* giảm hẳn so với ma trận $A.A^T$

- **Bước 6.2:** Tính các vector riêng V_i (eigenvectors) của ma trận vuông $A^T.A$ này.

Cụ thể ta có:

E =

-0.5000	-0.3435	0.7887	0.0994
-0.5000	-0.2488	-0.3294	-0.7613
-0.5000	-0.2716	-0.5159	0.6404
-0.5000	0.8639	0.0566	0.0216

F =

1.0e+04 *

-0.0000	0	0	0
0	3.8903	0	0
0	0	4.9698	0
0	0	0	8.9589

Ma trận E chứa các vector riêng, ma trận F chứa các trị riêng tương ứng. Ở đây ta tìm được 4 trị riêng của ma trận $A^T.A$, tuy nhiên ta sẽ sắp xếp lại theo thứ tự giảm dần, và chỉ lấy các trị riêng "non-zero". Kết quả ta được 3 trị riêng (từ đó tính ra 3 vector riêng tương ứng):

V_1 =	V_2 =	V_3 =
0.0994	0.7887	-0.3435
-0.7613	-0.3294	-0.2488
0.6404	-0.5159	-0.2716
0.0216	0.0566	0.8639
λ_1 = 89589	λ_2 = 49698	λ_3 = 38903

Đây là các *eigenvector* của $A^T.A$ tương ứng với các *eigenvalues* λ. Kích thước *Mx1*.

Sau khi đã tính được các vector V_i (có kích thước là *Mx1*), ta sẽ dễ dàng suy ra được các vector riêng U_i *(*kích thước N^2x1*)* mong muốn cần tìm, theo công thức:

$$U_i = AV_i \qquad (5)$$

Cụ thể ta có:

U_1 =	U_2 =	U_3 =
18.3984	-42.0673	71.7485
-106.7796	95.1990	69.0653
97.0356	-18.4503	19.9105
-118.7657	93.0798	51.2723
-4.1672	35.4840	-20.1623
-9.6605	-76.5643	32.4590
156.8702	-14.0823	66.1679
-107.4913	-113.5909	110.5418
134.3656	98.1768	88.8929

Đây là các eigenvector của ma trận hiệp phương sai C ($A^T.A$) cần phải tìm. Kích thước N^2x1

Cuối cùng ta chuẩn hóa các vector U_i :

$$U_i = \frac{U_i}{||U_i||} \qquad (6)$$

Sau khi chuẩn hóa ta thu được các vector U_i cuối cùng như sau:

U_1 =	U_2 =	U_3 =
0.0615	-0.1887	0.3638
-0.3567	0.4270	0.3502
0.3242	-0.0828	0.1009
-0.3968	0.4175	0.2600
-0.0139	0.1592	-0.1022
-0.0323	-0.3434	0.1646
0.5241	-0.0632	0.3355
-0.3591	-0.5095	0.5604
0.4489	0.4404	0.4507

Đây chính là các vector cơ sở của không gian mới. Ta gọi các vector này là các ***EIGENFACES.***

- **Bước 6.3:** Tính *M* vector riêng U_i tốt nhất của $A.A^T$ theo công thức (5)

Bước 7: Chỉ giữ lại *K* vector riêng trong số M vector nói trên (ứng với *K* trị riêng *lớn nhất*), *K rất nhỏ so với* N^2.

Có 2 cách để xác định *K*.

Cách 1:

- Sắp xếp theo thứ tự dãy *giảm dần* các eigenvalues tìm được.

- Theo dõi sự biến thiên của dãy trên, khi không còn biến thiên (hoặc xấp xỉ bằng không) thì lúc đó ta đã chọn đủ *K*.

Cách 2:

Chọn K theo công thức sau:

$$\frac{\sum_{i=1}^{k}\lambda_i}{\sum_{i=1}^{N}\lambda_i} > Ngưỡng\ (ví\ dụ\ 0.9\ hoặc\ 0.95) \quad (7)$$

Biểu diễn tập huấn luyện vào không gian vector riêng

Mỗi khuôn mặt Φ_i trong tập huấn luyện có thể được biểu diễn lại là 1 tổ hợp tuyến tính của K vector riêng lớn nhất:

$$\sum_{j=1}^{K} W_j U_j \qquad (8)$$

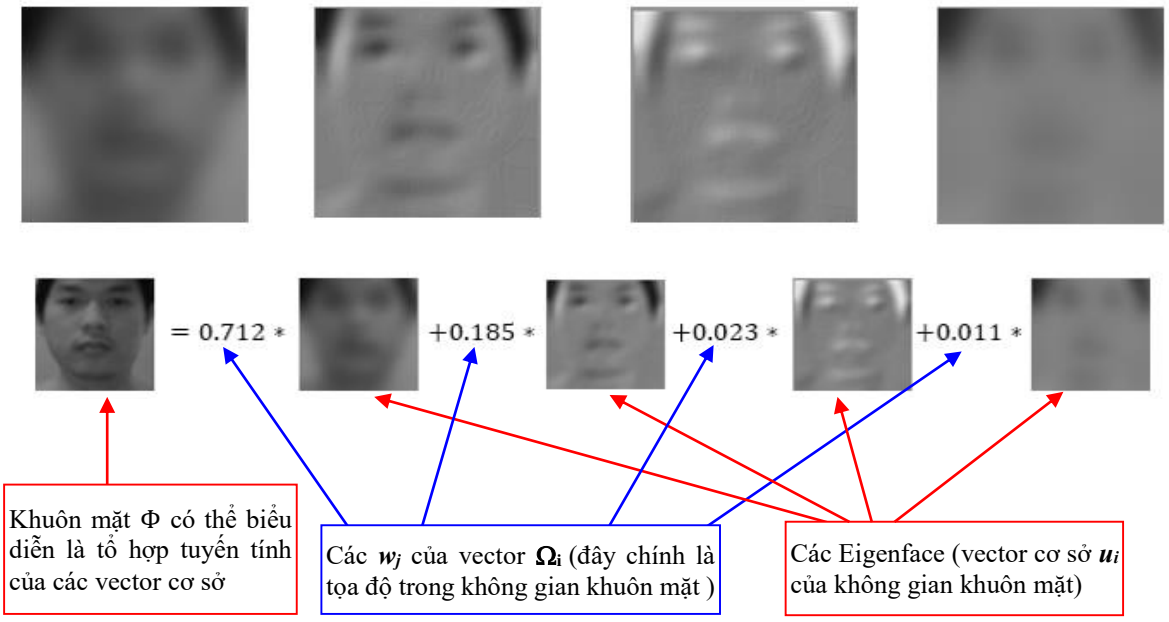
Trong đó:

$$W_j = U_j^T \Phi_i \qquad (9)$$

Gọi các ***"vector riêng U_i"*** là các **"EIGENFACES"** (khuôn mặt riêng). Như vậy, bây giờ, mỗi khuôn mặt huấn luyện Φ_i sẽ được biểu diễn trong không gian mới là 1 vector như sau:

$$\Omega_i = \begin{bmatrix} W_1^i \\ W_2^i \\ \dots \\ W_K^i \end{bmatrix}, \text{ i = 1,2,…,M} \qquad (10)$$

$$\Omega_i = \begin{bmatrix} U_1^T\Phi_i \\ U_2^T\Phi_i \\ \dots \\ U_K^T\Phi_i \end{bmatrix} \qquad (11)$$



Hình 2.2. *Minh họa các Eigenfaces[6]*

Nhận dạng khuôn mặt bằng các Eigenfaces

Cho 1 ảnh khuôn mặt chưa biết là ai (khuôn mặt giống tập mẫu – chính diện và cùng kích thước). Giai đoạn nhận dạng này giống hệt giai đoạn biểu diễn ảnh khuôn mặt trong tập mẫu. Ta thực hiện lần lượt các bước sau:

Bước 1: Chuẩn hóa Γ:

$$\Phi = \Gamma_i - \Psi \qquad (12)$$

Bước 2: Biểu diễn Φ thành Ω như sau:

$$\Omega_i = \begin{bmatrix} U_1^T \Phi_i \\ U_2^T \Phi_i \\ \dots \\ U_K^T \Phi_i \end{bmatrix} \qquad (13)$$

Bước 3: Tìm khuôn mặt thứ *L* trong tập mẫu có khoảng cách gần nhất với khuôn mặt cần nhận dạng.

$$e_r = min_l ||\Omega - \Omega_L|| \qquad (14)$$

Với Ω_L là vector mô tả khuôn mặt thuộc thứ L.

Bước 4: Phân loại và nhận dạng

Sau khi trích chọn được vector đặc tính, chúng ta cần đối chiếu vector này với cơ sở dữ liệu, từ đó đưa ra kết quả nhận dạng. Trong bài toán, kết quả nhận dạng sẽ là nhận biết được hoặc chưa nhận biết được.

Để thực hiện phân loại có rất nhiều phương pháp như khoảng cách Euclide, mạng Noron, SVM,… trong đó khoảng cách Euclide là phương pháp đơn giản nhất. Nó cho kết quả tốt đối với trường hợp đối tượng đã được tạo thành các nhóm cách xa nhau. Vector đặc tính của đối tượng cần nhận dạng sẽ được so sánh lần lượt với vector đặc tính của từng ảnh mẫu trong tập các ảnh học. Các khoảng cách ngắn nhất sẽ được lưu lại.

Nếu $e_r < T_r$ (T_r là 1 ngưỡng chấp nhận được nào đó):

Tức ảnh khuôn mặt cần xác định "đủ gần" với ảnh của người thứ *l* trong tập mẫu. Khi đó, ta kết luận đó chính là khuôn mặt của người thứ *l*.

Ở bài báo tác giả sử dụng phương pháp phân loại, nhận dạng là dùng khoảng cách Euclide. Khi chọn ngưỡng quá thấp thì độ

chính xác nhận dạng tăng nhưng số lượng ảnh không nhận dạng được cũng tăng, khi chọn ngưỡng quá cao thì số lượng ảnh nhận dạng được tăng lên nhưng tỷ lệ độ chính xác giảm đi, trong bài báo tác giả đã điều chỉnh ngưỡng để có tỷ lệ nhận dạng chính xác các ảnh trong tập huấn luyện là trên 90%.

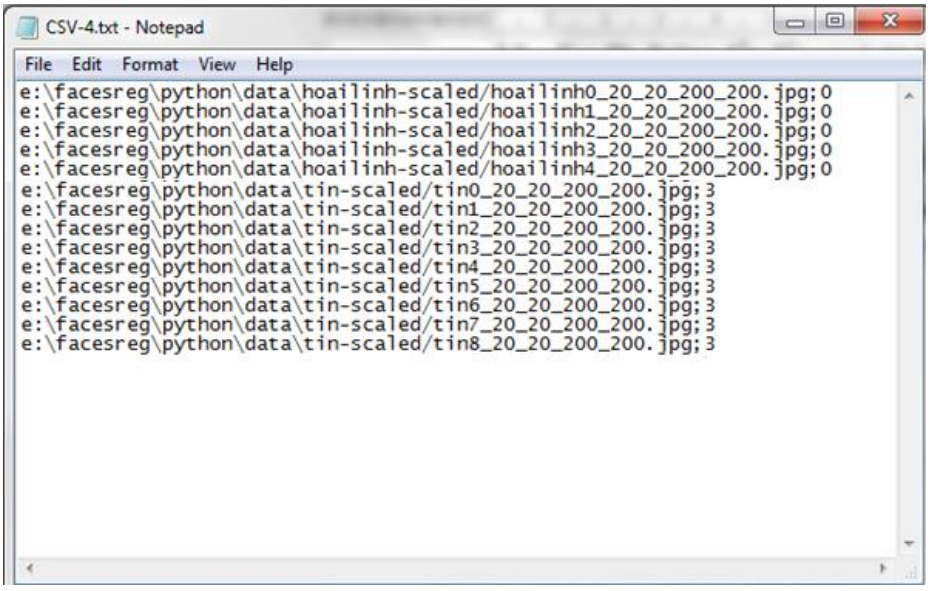
Bộ dữ liệu được chọn sẽ bao gồm 2 phân lớp. Phân lớp thứ nhất gồm 5 ảnh khuôn mặt của nghệ sĩ Hoài Linh, phân lớp thứ 2 gồm 9 ảnh khuôn mặt của tác giả. Tất cả các khuôn mặt được lấy mẫu trong điều kiện ánh sáng đầy đủ, và các mặt nghiên đã được xoay về mặt chính diện. Kích thước mỗi ảnh đều bằng nhau.

Để nhận dạng ảnh trực tiếp từ camera với các kích cỡ khuôn mặt khác nhau, trong bài báo này tác giả sử dụng 2 cơ sở dữ liệu về đặc trưng gương mặt người được huấn luyện theo thuật toán AdaBoost [7] được OpenCV [8] cung cấp là:

- Haarcascade_frontalface_alt2.xml: Đặc trưng gương mặt theo hướng nhìn chính diện [8].
- Haarcascade_profileface.xml: Đặc trưng gương mặt theo hướng nhìn từ cạnh bên [8].

Sau khi phát hiện được khuôn mặt sẽ tiến hành vẽ khung viền (bouding box), điều chỉnh kích thước khuôn mặt về cùng kích thước tập mẫu huấn luyện để nhận dạng và đặt nhãn text.

Để thuận tiện cho việc nhận dạng người mới, tác giả đặt các gương mặt huấn luyện ở cùng một phân lớp (một người) vào chung một thư mục và tạo file CSV.txt để lưu đường dẫn đến các ảnh trong thư mục đó. Khi có thêm người mới cần nhận dạng sẽ tiến hành lấy mẫu và tạo lại file CSV.txt

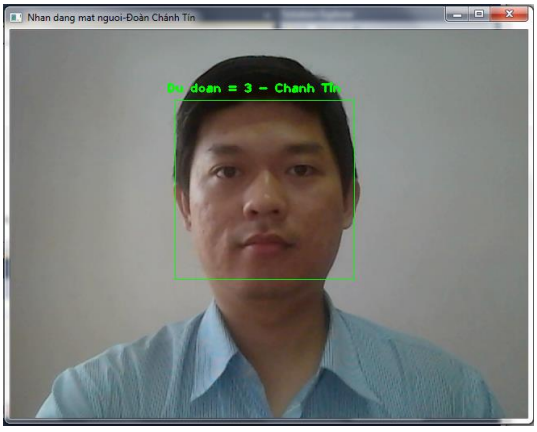


Hình 2.3. *File CSV.txt chứa đường dẫn đến các ảnh huấn luyện*

3. Kết quả

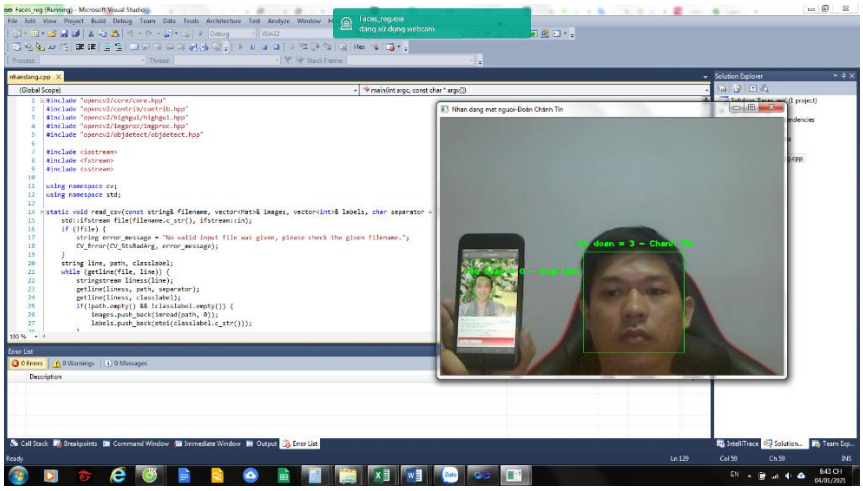
Chương trình được ứng dụng chạy thử nghiệm trong nhiều điều kiện ánh sáng khác nhau, từ môi trường có ánh sáng yếu cho đến môi trường đủ sáng ở các khoảng cách khác nhau, nhận dạng một hoặc nhiều người xuất hiện cùng lúc, và đạt được các kết quả như sau:

- Kết quả nhận dạng một người



Hình 3.1. *Ảnh nhận dạng tác giả - phân lớp 2*

- Kết quả nhận dạng hai người



Hình 3.2.*Nhận dạng khi xuất hiện mặt người ở phân lớp 1,2 và giao diện trên máy tính*

- Kết quả nhận dạng khi có người lạ xuất hiện



Hình 3.3. *Phân loại được khi có người lạ xuất hiện*



Hình 3.4. *Nhận dạng người đã huấn luyện và người lạ (chưa huấn luyện)*

4. Kết luận và hướng phát triển

Kết luận

- Trên cơ sở nghiên cứu lý thuyết tổng quan về các phương pháp nhận dạng mặt người trong thời gian thực. Đặc biệt là thuật toán PCA, các đặc trưng Eigenfaces và Eigenvector kết hợp với kỹ thuật lập trình C++, Python. Tác giả đã áp dụng thành công thuật toán nhận dạng phân tích thành phần chính PCA vào bài toán nhận dạng mặt người trong thời gian thực. Xây dựng thành công phần mềm nhận dạng được một hoặc nhiều người xuất hiện trong khung ảnh, đặc biệt chương trình xử lý ảnh động, lấy trực tiếp từ camera. Bên cạnh đó, trong quá trình xây dựng chương trình, tác giả đã tìm hiểu về thư viện mã nguồn mở OpenCV, qua đó biết cách sử dụng các hàm trong thư viện OpenCV để xây dựng chương trình nhận dạng mặt người trên kit nhúng sau này.

- Về chương trình demo, chương trình được chạy thử nghiệm trên máy tính có bộ xử lý Core i7, thế hệ 4, tốc độ 2,4GHz; RAM 8GB. Tốc độ xử lý của chương trình 24 khung hình/s, khi ép xung lên 3,2Ghz thì có thể xử lý được 40 khung hình/s. Hiệu suất của chương trình còn tùy theo chất lượng của loại webcam được sử dụng, tập cơ sở dữ liệu lấy mẫu, và điều kiện ánh sáng khi lấy mẫu. Trong điều kiện ánh sáng phù hợp với môi trường ánh sáng khi lấy mẫu thì chương trình cho kết quả nhận dạng tốt, nhưng khi điều kiện ánh sáng thay đổi thì chương trình hay cho kết quả lẫn lộn giữa các phân lớp. Điều này cho thấy thuật toán PCA rất dễ bị ảnh hưởng của nhiễu do gương mặt bị nghiêng, hay mặt có râu, mặt cười, và đặc biệt là nhiễu do ánh sáng.

Hướng phát triển

- Về cơ bản, bài báo đã hoàn thành các yêu cầu ban đầu đề ra. Tuy nhiên, để ứng dụng vào thực tế thì thuật toán PCA vẫn còn nhiều điểm hạn chế, mà cụ thể đó là ảnh hưởng từ nhiễu. Do đó, hướng phát triển tiếp theo của bài báo sẽ tập trung vào phần xử lý nhiễu, hoặc kết hợp với các thuật toán nhận dạng khác để xây dựng một hệ thống nhận dạng chính xác hơn ở những môi trường khác nhau, ở các góc nhìn gương mặt khác nhau, v.v. Từ đó áp dụng vào thực tế như hệ thống giám sát, kiểm tra an ninh, điểm danh học sinh tự động, kiểm tra một người có phải là cán bộ, giảng viên, công nhân viên của trường hay không, v.v.

- Kết nối với hệ thống Camera IP để có thể xây dựng 1 trung tâm giám sát an ninh cho công ty, chung cư, v.v.

- Tăng khả năng phát hiện của camera bằng cách sử dụng camera hồng ngoại hoặc sử dụng loại camera có độ phân giải cao.

- Kết hợp với các thuật toán nhận dạng khác để có tỷ lệ nhận dạng chính xác hơn.

TÀI LIỆU THAM KHẢO

[1] Trương Công Lợi, *Nhận dạng khuôn mặt sử dụng phương pháp biến đổi Eigenfaces và mạng Nơron,* NXB Đại học Đà Nẵng, 2013.

[2] Mạch Thị Kim Hạnh, *Xác thực sinh trắc học khuôn mặt trên thiết bị di động*, NXB Đại học Lạc Hồng, 2013.

[3] M. A. Turk và A. P. Pentland, "Face Recognition Using Eigenfaces", *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition,* 1991.

[4] Shruti Sehgal, Harpreet Singh, Mohit Agarwal, V. Bhasker, Shantanu, "Data analysis using principal component analysis", *Proc. of IEEE,* 2015.

[5] Jolliffe, I.T., *Principal Component Analysis,* Second edition (Springer), 2008.

[6] M. Turk and A. Pentland, "Eigenfaces for Recognition", *Journal of Cognitive Neuroscience,* 1991.

[7] Viola, P. and Jones, "Rapid object detection using a boosted cascade of simple features", *Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Dec,* 2001.

[8] *Face Detection using Haar Cascades*, <https://docs.opencv.org/3.4/d1/dee/tutorial_introduction_to_pca.html>

[9] Ahmed, N., Khan, F. A., Ullah, Z., Ahmed, H., Shahzad, T., & Ali, N., "Face Recognition Comparative Analysis Using Different Machine Learning Approaches", *Advances in Science and Technology Research Journal*, 15(1), 265-272, 2021.

[10] Bahurupi, S. P., & Chaudhari, D. S., "Principal component analysis for face recognition", *International Journal of Engineering and Advanced Technology*, 1(5), 91-94, 2012.

Ngày nhận bài: 27/01/2022 Biên tập xong: 14/11/2022 Duyệt đăng: 14/12/2022