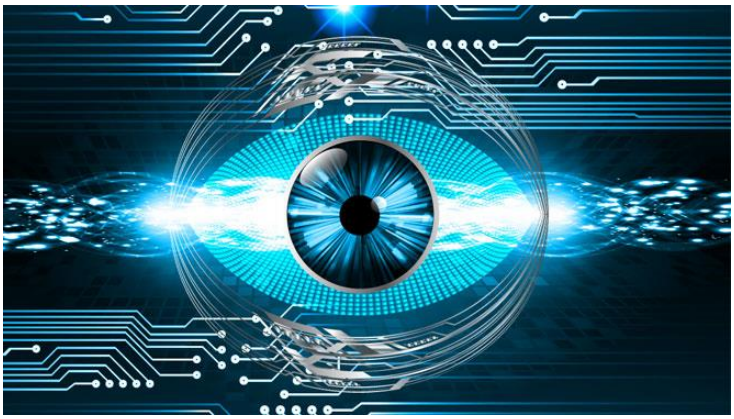
ĐỒ ÁN MÔN THỊ GIÁC MÁY TÍNH

ĐỀ TÀI :

NHẬN DẠNG MẶT NGƯỜI TRÊN MATLAB



Giảng viên hướng dẫn: Lê Thị Ngọc Thúy

Sinh viên thực hiện: Đỗ Thanh Huy

Trần Quốc Tuân

Đỗ Thị Thảo

Cù Quang Anh

MỤC LỤC

LỜI MỞ ĐẦU

Trong thế giới ngày nay với sự phát triển mạnh mẽ của kỹ thuật số và mạng toàn cầu, vấn đề về đảm bảo an toàn về thông tin cũng như vật chất ngày càng trở nên quan trọng và khó khăn. Một trong những phương pháp đơn giản nhất nhưng cũng là bài toán phức tạp nhất là xác định một người thông qua khuôn mặt từ đó cập nhật thông tin để đảm bảo an toàn thông tin cũng như tìm kiếm tội phạm.

Trong nhiều năm qua có rất nhiều công trình nghiên cứu về bài toán nhận dạng khuôn mặt người từ ảnh đen trắng, xám đến ảnh màu như ngày hôm nay. Các nghiên cứu đi từ bài toán đơn giản, mỗi ảnh chỉ có một khuôn mặt người nhìn thẳng vào thiết bị thu hình và đầu ở tư thế thẳng đứng trong ảnh đen trắng. Cho đến ngày hôm nay bài toán mở rộng cho ảnh màu, có nhiều khuôn mặt trong cùng một ảnh, có nhiều tư thế thay đổi trong ảnh. Không những vậy mà còn mở rộng phạm vi từ môi trường xung quanh khá đơn giản cho đến môi trường xung quanh rất phức tạp nhằm đáp ứng nhu cầu của con người.

Mục tiêu của đề tài "Nhận dạng mặt người bằng thuật toán PCA trên Matlab" là thực hiện 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.

Để tiện theo dõi em xin trình bày đề tài theo ba phần:

- ✓ Phần đầu là tổng quan về nhận diện khuôn mặt và thuật toán PCA sử dụng trong bài.
- ✓ Phần 2 giới thiệu về ảnh màu và những thứ liên quan đến việc nhận dạng khuôn mặt và phần mềm sử dụng.
- ✓ Sau cùng là code chương trình và thuật toán để dễ hình dung hơn

Do tài liệu tham khảo hạn chế, trình độ có hạn và kinh nghiệm thực tiễn còn non kém nên đề tài còn nhiều thiếu sót. Chúng em mong nhận được những ý kiến và lời nhận xét của cô để giúp chúng em hoàn thiện hơn đề tài này.

PHẦN I: TỔNG QUAN VỀ NHẬN DIỆN KHUÔN MẶT VÀ THUẬT TOÁN PCA

I. CÔNG NGHỆ NHẬN DIỆN MẶT NGƯỜI TRONG THỊ GIÁC MÁY TÍNH.

Sinh trắc học được sử dụng trong quá trình xác thực người bằng cách sử dụng các đặc trưng của con người để xác minh hoặc nhận dạng. Có rất nhiều loại hệ thống sinh trắc học như nhận diện dấu vân tay, nhận diện và phát hiện khuôn mặt, nhận diện mống mắt…Các đặc trưng sinh học này này thường được sử dụng cho nhận dạng người trong hệ thống giám sát hoặc nhận dạng tội pham. Lợi thế của việc sử dụng đặc trưng sinh học trong nhận dạng đó là chúng rất khó có thể thay đổi theo thời gian và là các đặc trưng độc nhất của mỗi người.

Một hệ thống nhận diện khuôn mặt là một ứng dụng máy tính, có khả năng nhận dạng và xác minh một người từ một ảnh số hoặc từ khung video trong video.

Các pha trong một hệ thống nhận diện khuôn mặt:

Để xây dựng một hệ thống nhận dạng mặt, cũng không hề đơn giản, bước đầu tiên cần thực hiện là face detection, tức là phát hiện phần ảnh mặt trong dữ liệu đầu vào (CSDL ảnh, video …) và cắt lấy phần ảnh mặt để thực hiện nhận dạng (Face cropping), bước thứ hai là tiền xử lý ảnh (Preprocessing) bao gồm các bước căn chỉnh ảnh (Face image alignment) và chuẩn hóa ánh sáng (Illumination normalization) (ở đây đang nói tới các ảnh có góc nhìn thẳng), tiếp đến là bước trích chọn đặc trưng (Feature extraction), ở bước này một phương pháp trích chọn đặc trưng nào đó (mẫu nhị phân cục bộ – Local Binary Pattern – LBP, Gabor wavelets, …) sẽ được sử dụng với ảnh khuôn mặt để trích xuất các thông tin đặc trưng cho ảnh, kết quả là mỗi ảnh sẽ được biểu diễn dưới dạng một vector đặc trưng, bước tiếp theo là bước nhận dạng hay phân lớp, tức là xác định danh tính hay nhãn của ảnh – đó là ảnh của ai. Ở bước phân lớp, thường thì phương pháp *k-nearest neighbor* sẽ được sử dụng, ngoài ra có thể sử dụng *SVM (Support Vector Machine)* tuy nhiên không mang lại hiệu quả cao. Dữ liệu cho một hệ thống nhận dạng mặt được chia làm 3 tập: tập huấn luyện (Training set), tập tham chiếu (reference set hay gallery set) và tập để nhận dạng (probe set hay query set, đôi khi còn gọi là test set). Trong nhiều hệ thống, tập training trùng với tập reference. Tập training gồm các ảnh được dùng để huấn luyện (hay học), thông thường tập này được dùng để sinh ra một không gian con (projection subspace) là một

ma trận và phương pháp hay được sử dụng là *PCA (Principal Component Analysis), WPCA (Whitened PCA), LDA (Linear Discriminant Analysis), KPCA (Kernel PCA).* Tập reference gồm các ảnh đã biết danh tính được chiếu (projected) vào không gian con ở bước training. Bước training nhằm 2 mục đích: giảm số chiều (Dimension reduction) của các vector đặc trưng (Feature vector) vì các vector này thường có độ dài khá lớn (vài nghìn tới vài trăm nghìn) nên nếu để nguyên thì việc tính toán sẽ rất rất lâu, thứ hai là làm tăng tính phân biệt giữa các ảnh khác lớp (định danh khác nhau), ngoài ra có thể làm giảm tính phân biệt giữa các ảnh thuộc về một lớp (tùy theo phương pháp, ví dụ như *Linear Discriminant Analysis* LDA- còn gọi là *Fisher Linear Discriminant Analysis-Fisherface* là một phương pháp làm việc với tập Training mà mỗi đối tượng có nhiều ảnh mặt ở các điều kiện khác nhau). Sau khi thực hiện chiếu tập Reference vào không gian con, hệ thống lưu lại kết quả là một ma trận với mỗi cột của ma trận là một vector tương ứng với ảnh (định danh đã biết) để thực hiện nhận dạng (hay phân lớp). Nhận dạng (hay phân lớp) được thực hiện với tập các ảnh khảo sát, sau khi tiền xử lý xong, mỗi ảnh sẽ được áp dụng phương pháp trích chọn đặc trưng (như với các ảnh thuộc tập training và Reference) và được chiếu vào không gian con. Tiếp đến việc phân lớp sẽ dựa trên phương pháp k-NN, định danh của một ảnh cần xác định sẽ được gán là định danh của ảnh có khoảng cách (distance) gần với nó nhất. Ở đây cần lưu ý là mỗi ảnh là một vector nên có thể dùng khái niệm hàm khoảng cách giữa hai vector để đo sự khác biệt giữa các ảnh.

Ứng dụng của nhận diện khuôn mặt người:

- Hệ thống tương tác giữa người và máy: giúp những người bị tật hoặc khiếm khuyết có thể trao đổi.
- Hệ thống quan sát, theo dõi và bảo vệ.
- Thẻ căn cước, chứng minh thư nhân dân (Face Indentification).
- Tìm kiếm và tổ chức 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.
- Ứng dụng trong video phone.
- Phân loại trong lưu trữ hình ảnh trong điện thoại di động.
- Kiểm tra trạng thái người lái xe có ngủ gật, mất tập trung hay không, và hỗ trợ thông báo khi cần thiết.
- Trong lĩnh vực thiết kế điều khiển robot.

Phương pháp nhận diện khuôn mặt người:

Dựa vào những đặc điểm của phương pháp nhận diện khuôn mặt người trên ảnh. Các phương pháp này được chia làm bốn hướng tiếp cận chính:

- Hướng tiếp cận dựa trên tri thức: Mã hóa các hiểu biết của con người về khuôn mặt thành các luật. Thông thường các luật mô tả mối quan hệ giữa các đặc trưng.
- Hướng tiếp cận dựa trên các đặc trưng không thay đổi: Mục tiêu của các thuật toán này là đi tìm các đặc trưng mô tả cấu trúc khuôn mặt người mà các đặc trưng này sẽ không thay đổi theo thời gian, và không phụ thuộc vào biểu cảm khuôn mặt, vị trí đặt thiết bị thu hình hay điều kiện ánh sáng.
- Hướng tiếp cận dựa trên so mẫu khớp: Dùng các mẫu chuẩn của khuôn mặt người (các mẫu này được lựa chọn và lưu trữ) để mô tả cho khuôn mặt người hay các đặc trưng khuôn mặt (các mẫu này phải được chọn làm sao cho tách biệt nhau theo tiêu chuẩn mà các tác giả định ra để so sánh). Các mối tương quan giữa những dữ liệu ảnh đưa vào và các mẫu dùng để xác định khuôn mặt người.
- Hướng tiếp cận dựa trên diện mạo: Trái ngược hẳn với so mẫu khớp, các mô hình (hay các mẫu) được học từ một tập ảnh huấn luyện trước đó. Sau đó hệ thống (mô hình) sẽ xác định khuôn mặt người. Hay một số tác giả còn gọi hướng tiếp cận này là hướng tiếp cận theo phương pháp học.

Ưu và nhược điểm

So với các công nghệ khác

Trong số các kỹ thuật sinh trắc học, nhận dạng khuôn mặt có thể không phải là phương pháp đáng tin cậy và hiệu quả nhất. Tuy nhiên, một trong những lợi thế quan trọng là nó không đòi hỏi sự hợp tác của các đối tượng thử nghiệm. Các hệ thống thiết kế được lắp đặt tại các sân bay, khu chung cư, và những nơi công cộng khác có thể xác định các cá nhân giữa đám đông, mà không bỏ sót một ai. Sinh trắc học khác như dấu vân tay, quét mống mắt, và nhận dạng giọng nói không thể thực hiện được điều này.

Nhược điểm

Nhận dạng khuôn mặt còn rất xa mới có thể đạt đến mức độ hoàn hảo, ngoài ra cũng rất khó để thực hiện phương pháp này trong các điều kiện nhất định. Ralph Gross, một nhà nghiên cứu tại Viện Mellon Robotics Carnegie, mô tả một trở ngại liên quan đến các góc nhìn của khuôn mặt: "Nhận dạng khuôn mặt đã thực hiện được khá tốt ở phía mặt trước và phía chênh lệch 20 độ, nhưng ngay sau khi bạn đi về phía góc khuất, thì nó có vấn đề."

Các điều kiện khác mà nhận dạng khuôn mặt không làm việc tốt bao gồm thiếu ánh sáng, đeo kính mát, tóc dài, hoặc các đối tượng mà một phần khuôn mặt bị che, và các hình ảnh độ phân giải thấp. Một bất lợi nghiêm trọng là nhiều hệ thống sẽ kém

hiệu quả nếu biểu hiện khuôn mặt khác nhau. Ngay cả một nụ cười lớn, cũng có thể làm cho hệ thống giảm tính hiệu quả. Ngoài ra còn có sự không thống nhất trong các bộ dữ liệu được sử dụng bởi các nhà nghiên cứu.

Thách thức đối với nhận diện khuôn mặt:

Hiện nay các vấn đề sau được coi là thách thức lớn (chưa có phương pháp tối ưu) đối với nhận dạng mặt:

- Sự thay đổi hướng khuôn mặt: kết quả với các ảnh có hướng thay đổi (> 45°, không phải chính diện) còn khá khiêm tốn.
- Độ phân giải thấp: ảnh thu được từ các camera giám sát thường có kích thước và chất lượng rất rất thấp, các kết quả nghiên cứu về lĩnh vực này còn chưa nhiều.
- Nhận diện khuôn mặt dựa trên video: với sự phát triển của các phương tiện multimedia, thông tin mặt người trong các dữ liệu video là vô cùng nhiều, tuy nhiên hầu hết các phương pháp nhận dạng vẫn làm việc với ảnh tĩnh trích xuất từ dữ liệu video, chưa có phương pháp tốt tận dụng hết ưu thế của dữ liệu video.
- Các hệ thống lớn: các cơ sở dữ liệu ảnh khuôn mặt được test bởi các nhà nghiên cứu còn khá nhỏ (vài trăm tới vài chục nghìn ảnh mặt), tuy nhiên trên thực tế các CSDL có thể rất lớn, ví dụ CSDL ảnh khuôn mặt của cảnh sát có thể chứa từ hàng triệu tới hơn 1 tỉ ảnh …
- Điều kiện lão hóa: việc nhận dạng ảnh mặt thay đổi theo thời gian thực sự vẫn còn là một vấn đề lớn ngay cả đối với khả năng nhận dạng của con người.
- Điều kiện sáng: là một trong những thách thức lớn nhất của nhận dạng mặt, chưa có phương pháp tốt cho các ảnh chụp ở điều kiện ngoài trời.

II. THUẬT TOÁN PCA.

Phân tính thành phần chính (Principal component analysis) hay còn gọi là PCA là một trong những kết quả đẹp từ việc áp dụng đại số tuyến tính. PCA được sử dụng nhiều trong các khuôn mẫu phân tích bởi vì nó là phương pháp không cần tham số và đơn giản cho việc trích xuất thông tin thích hợp từ các tập dữ liệu không rõ ràng. PCA cung cấp một hướng đi cho việc làm thế nào để hạn chế một tập dữ liệu phức tạp tới một tập dữ liệu với số chiều nhỏ hơn để hiện ra thông tin ẩn dưới tập dữ liệu không rõ ràng đó.

Thông thường để hiểu rõ một hiện tượng nào đó, ta thường đo lường một vài đại lượng trong hệ thống. Ta có thể không tính toán được điều gì đã xảy ra do dữ liệu đo được rất mù mờ, không rõ ràng, số lượng các biến đo lường có thể rất lớn và tại các

thời điểm dễ gây nhầm lẫn. Ví dụ như với một hệ thống đơn giản ta chỉ cần một vài đo lường để tính toán hệ thống đó, nhưng đó là với trường hợp ta đã có những kinh nghiệm trước đó về hệ thống, nếu không có kinh nghiệm về hệ thống, thường thì ta sẽ sử dụng những hiểu biết đã có để đo lường hệ thống này, nhưng những hiểu biết này không phù hợp với hệ thống nên các phép đo này sử dụng một mô hình nhiều chiều hơn thực tế để mô tả hệ thống, gây nên dư thừa và dữ liệu không rõ ràng. Thuật toán PCA rất hữu dụng trong những trường hợp này. Nó sẽ biến đổi tập dữ liệu đo lường dư thừa, nhiễu lớn về tập dữ liệu mà biểu diễn diễn tốt nhất (dễ quan sát nhất, dễ phân biệt nhất) hệ thống. Với một hệ thống liên tục và tuyến tính nếu ta đưa các đo lường vào một không gian vector nơi mà mỗi thể hiện của hệ thống được xem như một vector thuộc không gian vector đó thì nó sẽ là tổ hợp tuyến tính của các vector cơ sở của không gian vector đó (số các vector cơ sở là số chiều của không gian đó). Ta sẽ sử dụng một phép biến đổi tuyến tính để ánh xạ tập dữ liệu gốc vào tập dữ liệu mới và ta mong muốn rằng tập dữ liệu mới này sẽ giảm sự dư thừa và nhiễu. Nhiệm vụ của ta là tìm ra phép biến đổi này.

Một số đặc điểm của PCA:

- Giúp giảm số chiều của dữ liệu.
- Thay vì giữ lại các trục tọa độ của không gian cũ, PCA xây dựng một không gian mới ít chiều hơn, nhưng lại có khả năng biểu diễn dữ liệu tốt tương đương không gian cũ, nghĩa là đảm bảo *độ biến thiên* (variability) của dữ liệu trên mỗi chiều mới.
- Các trục tọa độ trong không gian mới là tổ hợp tuyến tính của không gian cũ, do đó về mặt ngữ nghĩa, PCA xây dựng đặc trưng mới dựa trên các đặc trưng đã quan sát được. Điểm hay là những đặc trưng này vẫn biểu diễn tốt dữ liệu ban đầu.
- Trong không gian mới, các liên kết tiềm ẩn của dữ liệu có thể được khám phá, mà nếu đặt trong không gian cũ thì khó phát hiện hơn, hoặc những liên kết như thế không thể hiện rõ.

Giả sử, với X là ma trận các vector biểu diễn tập dữ liệu gốc, Y là ma trận các vector biểu diễn lại tập dữ liệu. P là phép biến đổi tuyến tính. Công thức để ánh xạ tập dữ liệu gốc vào tập dữ liệu biểu diễn lại như sau:

$$PX = Y \qquad (1)$$

Sau đây ta sẽ phân tích làm thế nào để xác định P. Có hai đại lượng toán học mà chúng ta cần quan tâm đó là phương sai và hiệp phương sai. Trong lý thuyết xác suất và thống kê:

- Phương sai của một biến ngẫu nhiên là một độ đo sự phân tán thống kê của biến đó, nó hàm ý các giá trị của biến đó thường ở cách giá trị kỳ vọng bao xa.

$$\sigma^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)}$$

- Hiệp phương sai là độ đo sự biến thiên cùng nhau của hai biến ngẫu nhiên.

$$cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})\,(Y_i - \bar{Y})}{(n-1)}$$

Hiệp phương sai của hai biến ngẫu nhiên X và Y cho biết mối tương quan giữa X và Y. Giá trị của hiệp phương sai không quan trọng bằng dấu của nó.

- Nếu giá trị hiệp phương sai là dương chỉ ra rằng X và Y tăng hoặc giảm cùng nhau.
- Nếu giá trị hiệp phương sai là âm sẽ chỉ ra rằng X sẽ tăng trong khi Y sẽ giảm hoặc ngược lại.
- Nếu giá trị hiệp phương sai bằng không, X và Y độc lập với nhau.

Hiệp phương sai là công cụ hữu dụng để tìm mối liên hệ giữa các chiều trong một tập dữ liệu có số chiều cao. Hiệp phương sai của một biến bằng phương sai của biến đó.

Các vector trong không gian vector được xem như một biến ngẫu nhiên nhiều chiều, mỗi thành phần của vector là một biến ngẫu nhiên vô hướng. Giả sử ta có vector sau.

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix}$$

Ma trân hiệp phương sai của của X có dạng sau.

$$C = \frac{1}{n-1}XX^T = \begin{bmatrix} cov(X_1,X_1) & cov(X_1,X_2) & \cdots & cov(X_1,X_n) \\ cov(X_2,X_1) & cov(X_2,X_2) & \cdots & cov(X_2,X_n) \\ \vdots & \vdots & \ddots & \vdots \\ cov(X_n,X_1) & cov(X_n,X_2) & \cdots & cov(X_n,X_n) \end{bmatrix}$$

Ta có nhận xét sau về ma trận hiệp phương sai:

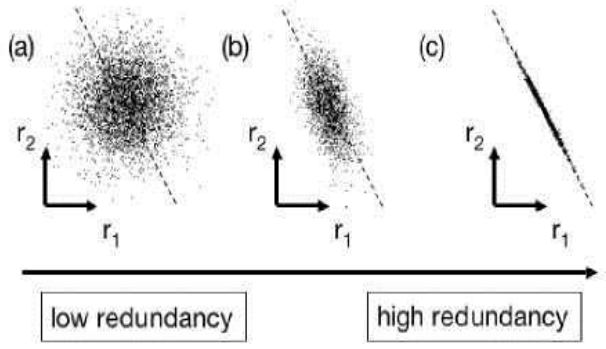
- Đường chéo ma trận là các phương sai của các thành phần của vector ngẫu nhiên X.
- Các vị trí không phải đường chéo cho thấy mối tương quan dữ liệu theo các chiều tương ứng trong vector ngẫu nhiên X.
- Ma trận C là ma trận đối xứng.

X là vector ngẫu nhiên nên ta có thể nói các thành phần không phải đường chéo của ma trận C biểu diễn mối tương quan giữa các chiều trong tập dữ liệu. Các thành phần đường chéo biểu diễn khả năng phân tán thống kê của dữ liệu theo chiều tương ứng. Như vậy để hệ thống có thể quan sát và phân biệt rõ ta cần tăng giá trị phương sai và giảm giá trị hiệp phương sai để giảm nhiễu và dư thừa. Nguyên nhân là do để biểu diễn tốt nhất tập dữ liệu thì nhiễu và dư thừa phải nhỏ nhất.

Một đo lường phổ biến cho biết chất lượng dữ liệu là tỉ số tín hiệu trên nhiễu SNR, hay tỉ số của phương sai σ^2.

$$SNR = \frac{\sigma^2_{tín\ hiệu}}{\sigma^2_{nhiễu}}$$

SNR cao ($\gg$1) chỉ ra rằng dữ liệu có độ chính xác cao, trong khi SNR thấp chỉ ra dữ liệu nhiễm nhiễu lớn. Trong ma trận hiệp phương sai các thành phần đường chéo là phương sai dữ liệu theo chiều tương ứng. Chiều có phương sai lớn tương ứng với hệ thống thay đổi rất nhiều với chiều này tức nghĩa là nó phụ thuộc rất nhiều vào những chiều này nên ta có thể coi đây là các thành phần $\sigma_{tín\ hiệu}$, còn chiều có phương sai nhỏ thì hệ thống thay đổi rất ít theo chiều đó nên ta có thể coi như $\sigma_{nhiễu}$ và cần loại bỏ (do những thành phần nhiễu thường thay đổi rất ít xung quanh một hằng số nào đó), và ta cũng có xu hướng loại bỏ các chiều có tính chất hằng do các chiều này không giúp phân biệt rõ giữa các đo lường trong hệ thống. Mục tiêu của chúng ta là tìm ra các chiều mới giúp phân biệt rõ các đo lường trong hệ thống. Những chiều không giúp phân biệt đưa vào hệ thống càng làm cho hệ thống khó hiểu hơn. Ta xét tiếp hình sau



Trên đây là một hệ tọa độ hai chiều r_1 và r_2, các chấm đen là các điểm trong hệ tọa độ và biểu diễn vị trí một vật nào đó trong hệ tọa độ 2 chiều, đường nét đứt là đường thẳng $r_2 = kr_1 + b$. Hình a). biểu diễn mức độ dư thừa trong tập quan sát thấp nhất. Hình c). biểu diễn mức độ dư thừa trong tập quan sát là nhiều nhất. Trong hình c) ta nhận thấy vị trí của vật gần như nằm trên đường thẳng, như vậy thay vì biểu diễn vị trí của vật trong hình c) bằng hệ tọa độ 2 chiều thì ta chỉ cần biểu diễn vị trí theo một trục tọa độ song song với đường thẳng $r_2 = kr_1 + b$. Việc biểu diễn bằng hai trục tọa độ ở đây là dư thừa và không cần thiết. Như vậy nếu dữ liệu biểu diễn theo 2 chiều mà phụ thuộc tuyến tính vào nhau thì sẽ xảy ra tình trạng dư thừa. Để tránh tình trạng dư thừa trong biểu diễn dữ liệu thì thì $cov(r_1, r_2) = 0$ hay mối tương quan giữa các chiều là bằng không.

Như vậy để biểu diễn tốt hệ thống thì ma trận hiệp phương sai của vector dữ liệu phải có các thành phần đường chéo có giá trị khác không và các thành phần không thuộc đường chéo có giá trị bằng không, nói cách khác ma trận hiệp phương sai phải là *ma trận chéo*. Nhắc đến ma trận chéo làm ta liên tưởng tới các bài toán chéo hóa ma trận trong đại số tuyến tính.

Trở lại với công thức (1). Ma trận hiệp phương sai của Y có dạng.

$$\begin{aligned} S_Y &= \frac{1}{n-1} YY^T \\ &= \frac{1}{n-1} (PX)(PX)^T \\ &= \frac{1}{n-1} PXX^T P^T \\ &= \frac{1}{n-1} P(XX^T)P^T \end{aligned}$$

$S_Y = \frac{1}{n-1} PAP^T$, Ở đây $A = XX^T$ và là ma trận đối xứng. $\frac{1}{n-1} A$ là ma trận hiệp phương sai của X.

Như vậy ta phải tìm P để S_Y là ma trân chéo. Từ công thức trên ta thấy bài toán trở về các bài toán chéo hóa trong đai số tuyến tính – Tìm ma trận P để PAP^T là ma trân chéo. Trong đại số tuyến tính, một ma trận đối xứng chéo hóa được bởi một ma trận các *vector riêng* trực giao của nó. Như vậy P phải là một ma trận trực giao với các hàng là các vector riêng của A. Ta có định nghĩa về vector riêng và giá trị riêng như sau:

λ được gọi là giá trị riêng của ma trân $A = [a_{ij}]_{n\times n}$ nếu tồn tại $x_1, \dots, x_n$ không đồng thời bằng 0 sao cho:

$$A\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

Khi đó $v = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^n$ *được gọi là vector riêng ứng với giá trị riêng* λ.

Trong đại số tuyến tính người ta đã chứng minh được rằng ma trận chéo hóa từ ma trận *A* có các thành phần đường chéo là các giá trị riêng của *A*. Như vậy ma trận hiệp phương sai của *Y* có các thành phần đường chéo là các giá trị riêng của *A* tương ứng với các vector hàng của *P* là các vector riêng của *A*. Ta mong muốn thành phần đường chéo của ma trận hiệp phương sai của *Y* càng lớn càng tốt. Như vậy ta chỉ lựa chọn các giá trị riêng lớn nhất và các vector riêng tương ứng với giá trị riêng đó cho thành phần đường chéo của S_Y và ma trận *P*.

Nói tóm lại, phương pháp PCA quy về việc đi tìm trị riêng (eigenvalues) và vector riêng (eigenvectors) 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.

III. THUẬT TOÁN PCA TRONG NHẬN DIỆN MẶT NGƯỜI

PCA được phát minh năm 1901 bởi Karl Pearson, là một thủ tục han chế biến khi thu được một tập dữ liệu có sự dư thừa. Thuật toán này giúp cho việc hạn chế các biến vào một số lượng nhỏ hơn các biến khác được gọi là các *thành phần chính*. Khi thực hiện nhận diện ảnh số lượng ảnh và kích thước ảnh là một vấn đề lớn trong không gian nhiều chiều. Mục tiêu của PCA trong nhận diện khuôn mặt là hạn chế chiều của dữ liệu bằng cách dữ lại càng nhiều sự biến thiên có thể càng tốt trong tập dữ liệu gốc (điều này nó thể làm mất thông tin). Không gian hạn chế chiều tốt nhất sẽ được xác định bởi các thành phần chính tốt nhất.

Nhận diện khuôn mặt sử dụng PCA trong phương pháp khuôn mặt riêng, giúp cho vệc giảm kích thước của cơ sở dữ liệu cho nhận diện ảnh thông qua loại bỏ các vector riêng bằng không và các vector riêng ứng với các giá trị riêng rất nhỏ, đồng thời làm nổi bật nét đặc trưng ảnh, tăng khả năng phân biệt ảnh do việc giảm mối tương quan giữa các chiều trong tập dữ liệu ảnh huấn luyện, hay nói cách khác làm căng không gian vector ảnh huấn luyện, nhờ đó mà bằng cách tính khoảng cách ngắn nhất với ảnh cần nhận dạng ta có thể dễ dạng tìm được ảnh tương tự trong cơ sở dữ liệu.

Trong phương pháp này ảnh được lưu trữ như các vector đặc trưng trong cơ sở dữ liệu, được thực hiện bằng cách ánh xạ mỗi ảnh huấn luyện vào không gian khuôn mặt riêng. Các đặc trưng này có thể có hoặc không liên quan với các đặc trưng thuộc về khuôn mặt như mắt, mũi, miệng hay tóc. Ảnh nhận diện được chiếu vào không gian khuôn mặt riêng và so sánh với ảnh trong cơ sở dữ liệu.

Phương pháp khuôn mặt riêng là một phương pháp hiệu quả được sử dụng trong nhận diện khuôn mặt do tính đơn giản, tốc độ nhanh và khả năng học của nó. Các khuôn mặt riêng là các thành phần chính của một sự phân bố khuôn mặt, hay nói cách khác, nó là các vector riêng của ma trận hiệp phương sai của ảnh khuôn mặt nơi mà mỗi ảnh khuôn mặt $N \times N$ pixel được coi như một điểm trong không gian N^2 chiều. Mỗi ảnh huấn luyện tương ứng với một vector riêng, cũng như ta có thể coi các vector riêng như một loại khuôn mặt. Mỗi ảnh khuôn mặt có thể biểu diễn chính xác bởi tổ hợp tuyến tính của các khuôn mặt riêng. Số lương có thể của các khuôn mặt riêng bằng với số lượng ảnh khuôn mặt trong tập huấn hiện. Các khuôn mặt cũng có thể biểu diễn gần đúng bằng cách sử dụng khuôn mặt riêng tốt nhất, cái mà tương ứng với các giá trị riêng lớn nhất, đại diện cho phương sai lớn nhất giữa tập các ảnh.

Các bước nhận diện ảnh sử dụng PCA.

1. Biểu diễn lại ảnh khuôn mặt

Tập huấn luyện với m ảnh kích thước $M \times N$ được biểu diễn lại thành các vector kích thước $M \times N$. Để đảm bảo tính chính xác thì mỗi khuôn mặt được chụp nhiều ảnh ở các góc độ khác nhau để tăng tính chính xác.

Mỗi khuôn mặt được ký hiệu bởi $\Gamma_1, \Gamma_2, \ldots \Gamma_m$.

Ví dụ:
$$\begin{bmatrix} 1 & -1 \\ 3 & 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ -1 \\ 2 \end{bmatrix}$$

Mỗi ảnh được biểu diễn lại như sau:

$$\Gamma_1 = \begin{bmatrix} 1 \\ -2 \\ 1 \\ -3 \end{bmatrix} \qquad \Gamma_2 = \begin{bmatrix} 1 \\ 3 \\ -1 \\ 2 \end{bmatrix} \qquad \ldots \qquad \Gamma_m = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 1 \end{bmatrix}$$

Trong Matlab hàm reshape() được sử dụng để thay đổi kích thước ma trận, cú pháp như sau:

a = reshape (b, m, n);

Với b là ma trận cần thay đổi, m, n là số hàng và cột của ma trận mới. m, n được lựa chọn sao cho số thành phần của ma trận gốc phải bằng với số thành phần của ma trận thay đổi kích thước.

2. Tìm ảnh trung bình và ảnh trung bình chuẩn hóa

Ảnh khuôn mặt trung bình là trung bình cộng các ảnh được tính bởi công thức:

$$\psi = \frac{1}{m}\sum_{i=1}^{m}\Gamma_i = (\Gamma_1 + \Gamma_2 + \cdots + \Gamma_m)/m$$

Ví dụ:

$$\begin{bmatrix}1\\-2\\1\\-3\end{bmatrix} + \begin{bmatrix}1\\3\\-1\\2\end{bmatrix} + \cdots + \begin{bmatrix}1\\2\\2\\1\end{bmatrix} \rightarrow \begin{bmatrix}-1\\-1\\2\\-3\end{bmatrix}$$

Ảnh trung bình chuẩn hóa là sự khác biệt giữa ảnh gốc và ảnh trung bình:

$$\phi_i = \Gamma_i - \psi$$

$$\phi_1 = \begin{bmatrix}2\\-1\\-1\\0\end{bmatrix} \qquad \phi_2 = \begin{bmatrix}2\\4\\-3\\5\end{bmatrix} \qquad \cdots \qquad \phi_m = \begin{bmatrix}2\\3\\0\\4\end{bmatrix}$$

Đoạn code sau sẽ lấy giá trị trung bình các cột của ma trận y và tính ϕ_i

```
t=mean(y,2);          % Tính ψ, các cột của y tương ứng Γi
for j=1:a
   y(:,j)=y(:,j)-t;   % Tính ϕ
end
```

Việc lấy ảnh trung bình chuẩn hóa giúp cho việc tính ma trận hiệp phương sai ở bước sau dễ dàng hơn

3. Tính ma trận hiệp phương sai

Ta coi mỗi ảnh tương ứng với một thể hiện của một lần chụp ảnh hay tương ứng với một đo lường. Vector ảnh là một vector ngẫu nhiên và mỗi một lần chụp là một thể hiển của vector ngẫu nhiên đó. Giả sử ta có A là một ma trận với các cột là các vector ϕ_i. Như vậy để tính mối tương quan giữa các chiều của vector ảnh ngẫu nhiên ta co công thức tính ma trận hiệp phương sai sau.

$C = AA^T$. A là một ma trận với các cột là các vector ϕ_i.

$A = [\phi_1, \phi_2, \dots, \phi_m]$

A có $M \times N$ hàng và m cột, do đó ma trận C có kích thước $(M \times N) \times (M \times N)$ rất lớn gây khó khăn cho các tính toán phía sau. Một cách đơn giản hơn là tính $A^T A$, ma trận này chỉ có kích thước $m \times m$. Nhưng ta cần tính toán các vector riêng của ma trận C nên ta sẽ biến đổi công thức như sau:

$$A^T A v_i = \lambda_i v_i$$

$$AA^T A v_i = A\lambda_i v_i$$

$$AA^T (A v_i) = \lambda_i (A v_i)$$

Như vậy các vector riêng của ma trận C bằng Av_i với v_i là vector riêng của ma trận $A^T A$. Do đó để tính các vector riêng của ma trận C có kích thước $(M \times N) \times (M \times N)$ đầu tiên ta tính các vector riêng của ma trận $A^T A$ sau đó nhân trái ma trận A, Av_i. Điều này giúp đơn giản trong việc tính toán.

4. Không gian khuôn mặt riêng

Các vector riêng của ma trận AA^T là Av_i được kí hiệu là U_i. Khi ta biến đổi lại kích thước của U_i ngược lại thành kích thước ảnh, nó giống như một ảnh khuôn mặt nhưng khó nhìn hơn và được gọi là các khuôn mặt riêng. Mỗi vector riêng ứng với một khuôn mặt riêng trong không gian khuôn mặt, các vector riêng bằng vector 0 được loại bỏ do đó cũng làm hạn chế không gian khuôn mặt tới một mức độ nào đó. Một ảnh khuôn mặt có thể được chiếu vào không gian khuôn mặt riêng bằng công thức.

$$\Omega = U^T \phi$$

Trong đó U là ma trận chuyển đổi có các cột là các vector riêng Av_i đã tìm được. ϕ là ảnh trung bình chuẩn hóa của ảnh khuôn mặt cần chiếu.

Trong matlab ta có hàm *eig* trả về giá trị riêng và vector riêng của một ma trận nào đó. Cú pháp như sau *[b c] = eig(A)* trong đó b là một ma trận với các cột là vector riêng, c là một ma trận với các thành phần đường chéo là các giá trị riêng tương ứng với các vector riêng trong b.

Để có thể chéo hóa trực giao được ma trận đối xứng thì ta cần trực chuẩn hóa hệ các vector riêng nhận được. Trong matlab có hàm *orth* trả về ma trận là hệ các vector trực chuẩn với mỗi vector tương ứng với các cột của ma trận.

5. Bước nhận diện ảnh

Ảnh kiểm tra Γ được chiếu vào không gian khuôn mặt để thu được vector Ω

$$\Omega = U^T(\Gamma - \psi)$$

Khoảng cách từ vector Ω tới mỗi vector khuôn mặt riêng Ω_k được gọi là khoảng cách Ơ-clit và được định nghĩa bởi công thức: $\epsilon_k^2 = \|\Omega - \Omega_k\|^2$ $k = 1,2,\ldots,m$

Trong đó Ω_k là một vector mô tả lớp khuôn mặt thứ *k*. Một khuôn mặt được xác định là thuộc lớp *k* nếu ϵ_k nhỏ nhỏ nhất và dưới một mức ngưỡng *T* chọn trước. Ngược lại được coi là không thuộc lớp *k*.

Trong trường hợp các ảnh tương đối giống nhau, vẫn có một số ảnh không phải ảnh cần nhận diện lọt dưới ngưỡng, khi đó có thể gây nhầm lẫn trong quá trình nhận diện. Để loại bỏ điều này ta sử dụng thuật toán láng giềng gần nhất. Ta sẽ xem xét khoảng cách từ ảnh cần nhận dạng tới mỗi ảnh mà dưới ngưỡng. Do mỗi khuôn mặt ta chụp nhiều ảnh, nên có thể coi mỗi khuôn mặt là một lop. Ta đếm số ảnh dưới ngưỡng cho mỗi lớp, ảnh thuộc lớp nào mà dưới ngưỡng nhiều nhất thì ảnh cần nhận dạng sẽ thuộc lớp đó.

PHẦN II: ẢNH MÀU VÀ
CÁC LỆNH XỬ LÝ ẢNH MÀU TRONG MATLAB

I. GIỚI THIỆU ẢNH SỐ

1. Ảnh số

Ảnh số là tập hợp các điểm ảnh với mức xám phù hợp dùng để mô tả ảnh gần với ảnh thật. Ảnh là một sự vật đại diện cho con người, sinh vật hay sự vật nào đó... Ảnh động như ta thấy trên truyền hình thực chất là tập hợp của rất nhiều ảnh tĩnh liên tiếp. Khi một ảnh đã được số hóa thì nó trở thành ảnh só và ảnh số này lại là một tập hợp của rất nhiều phần tử ảnh được gọi là điểm ảnh hay là "pixel". Mỗi điểm ảnh lại được biểu diễn dưới dạng một số hữu hạn các bit.

Chúng ta có thể chia ảnh ra làm ba loại khác nhau:

- *Ảnh đen trắng:* mỗi điểm ảnh được biểu diễn bởi một bit.
- *Ảnh Gray-scale:* mỗi điểm ảnh được biểu diễn bằng các mức chói khác nhau, thường thì ảnh này được biểu diễn bằng 256 mức chói hay là 8 bit cho mỗi điểm ảnh.
- *Ảnh màu:* mỗi điểm ảnh chia ra thành tín hiệu chói và tín hiệu màu.

Ảnh màu

1.1. Biểu diễn ảnh số

Trong biểu diễn ảnh, người ta thường áp dụng các phần tử đặc trưng của ảnh là Pixel. Nhìn chung có thể xem một hàm 2 biến chứa các thông tin biểu diễn của một ảnh. Các mô hình biểu diễn ảnh cho ta ta một mô tả logic hay định lượng các tính chất của hàm này.

Việc xử lý ảnh số phải được lấy mẫu và lượng tử hóa. Việc lượng tử hóa là chuyển đổi tín hiệu tương tự sang tín hiệu số của một ảnh đã lấy mẫu trong mọt số hữu hạn mức xám. Một số mô hình thường áp dụng biểu diễn ảnh: mô hình toán, mô hình thống kê

1.2. Ảnh màu

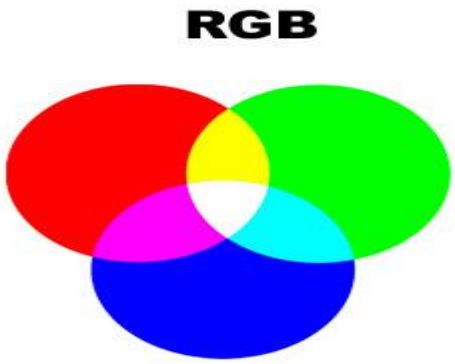
Cơ sở về ảnh màu

Như ta đã biết thì khi cho ảnh sáng trắng đi qua lăng kính ta sẽ thu được một dãy phổ màu bao gồm 6 màu rộng : tím lam, lục, vàng, cam , đỏ. Nếu nhìn kỹ thì sẽ không có ranh giới rõ ràng giữa các màu mà màu này sẽ từ từ chuyển sang màu kia. Mắt chúng ta nhìn thấy được là do ánh sáng phản xạ từ vật thể.

Tất cả các màu được tạo ra từ 3 màu cơ bản (màu sơ cấp) là: đỏ (R), lam (B), lục (G). Các màu cơ bản trộn lại với nhau theo một tỉ lệ nhất định để tạo ra các màu thứ cấp.

Phương trình màu:

$$Y = 0.2989 * R + 0.58662 * G + 0.11448 * B$$



Hình 3.2: Các màu cơ sở

Ví dụ: Đỏ + Lục = Vàng

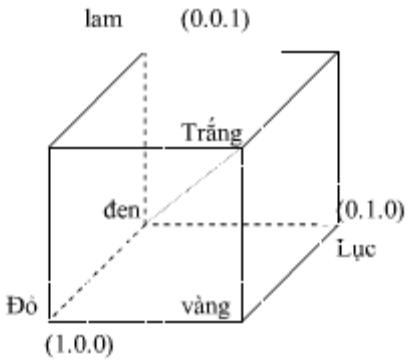
Lục + Lam = Xanh

Trộn ba màu sơ cấp hoặc trộn một màu thứ cấp với màu sơ cấp ngược với nó sẽ tạo ra được ánh sáng trắng.

Các màu gốc có liên quan đến các khái niệm sinh học hơn là vật lý, nó dựa trên cơ sở phản ứng sinh lý học của mắt người đối với ánh sáng. Mắt người có các tế bào cảm quang có hình nón nên còn được gọi là tế bào hình nón, các tế bào này thông thường có phản ứng cực đại với ánh sáng vàng-xanh lá cây (tế bào hình nón L), xanh lá cây (tế bào hình nón M), và xanh lam (tế bào hình nón S) tương ứng với các bước sóng khoảng 564 nm, 534 nm và 420 nm.

Các đặc trưng dùng để phân biệt một màu với màu khác là: độ sáng (brightness), sắc màu (hue) và độ bão hòa màu (sarturation).

- Màu sắc có liên quan đến bước sóng ánh sáng. Thông thường, sắc màu chính là tên của màu. Ví dụ: đỏ, cam, lục...
- Độ sáng thể hiện về cường độ ánh sáng: mô tả nó sáng hay tối như thế nào.
- Độ bão hòa màu: thể hiện độ thuần khiết của màu. Khi độ bão hòa cao, màu vẽ sạch và rực rỡ. Có nhiều mô hình màu như RGB, CYM, YIQ, CIE... Ở đây chỉ trình bày về mô hình màu RGB.



Hình 3.3: Mô hình màu RGB

Các màu R, G, B nằm ở các đỉnh trên trục tọa độ của khối vuông. Màu đen nằm ở gốc tọa độ, màu trắng nằm ở góc xa nhất so với điểm gốc. Thang màu xám kéo dài từ đen đến trắng (đường chấm).

Hình ảnh trong mô hình màu RGB bao gồm 3 mặt phẳng ảnh độc lập (dùng cho các màu sơ cấp).

Thường thì ta giả thiết là tất cả các giả trị màu được chuẩn hóa (tức là khối vuông là khối đơn vị), tất cả các giá trị màu nằm trong khoảng [0, 1]

Vì vậy trong hệ màu RGB, các màu có thể mô tả như là những điểm bên trong hình lập phương. Ở gốc tọa độ (0, 0, 0) là màu đen. Trên các trục tọa độ dương là các màu đỏ, lục, lam. Khi đó ánh sáng từ các điểm riêng biệt sẽ được cộng dồn với nhau để tạo ra các màu khác nhau.

(0, 0, 0)	màu đen
(255, 255, 255)	màu trắng
(255, 0, 0)	màu đỏ
(0, 255, 0)	màu xanh lá cây
(0, 0, 255)	màu xanh lam
(255, 255, 0)	màu vàng
(0, 255, 255)	màu xanh ngọc
(255, 0, 255)	màu hồng sẫm

1.3. Các định dạng ảnh cơ bản trong xử lý ảnh

- *Định dạng ảnh IMG:* là ảnh đen trắng, phần đầu của IMG có 16 byte chứa thông tin.

- *Định dạng ảnh GIF:* GIF (viết tắt của Graphics Interchange Format; trong tiếng anh là "Định dạng trao đổi hình ảnh") là một định dạng tập tin hình ảnh bitmap cho các hình ảnh dùng ít hơn 256 màu sắc khác nhau và các hoạt hình dùng ít hơn 256 màu cho mỗi khung hình. GIF là định dạng nén dữ liệu đặc biệt hữu ích cho việc truyền hình ảnh qua đường truyền lưu lượng nhỏ. Định dạng này được CopuServe cho ra đời năm 1987 và nhanh chóng được dùng rộng rãi trên World Wide Web cho đến nay. Tập tin GIF dùng nén dữ liệu bảo toàn trong đó kích thước tập tin có thể được giảm mà không làm giảm chất lượng hình ảnh, cho những hình ảnh có ít hơn 256 màu. Số lượng tối đa 256 màu làm cho định dạng này không phù hợp cho các hình chụp (thường có nhiều màu sắc), tuy nhiên các kiểu nén dữ liệu bảo toàn cho hình chụp nhiều màu cũng có kích thước quá lớn đối với truyền dữ liệu trên mạng hiện nay. Định dạng JPEG là nén dữ liệu thất thoát có thể được dùng cho các ảnh chụp, nhưng lại làm giảm chất lượng cho các bức vẽ ít màu, tạo nên những chỗ nhòe thay cho các đường sắc nét, đồng thopwif độ nén cũng thấp cho các hình vẽ ít màu. Như vậy GIF thường được dùng cho sơ đồ, hình vẽ nút bấm và các hình ít màu, còn JPEG được dùng cho ảnh chụp. Định dạng GIF dựa vào các bảng màu - một bảng chứa tối đa 256 màu khác nhau cho biết các màu được dùng trong hình

Hình 3.4: Ảnh GIF

- *Định dạng JPEG:* Phương pháp nén ảnh JPEG (tiếng Anh, viết tắt cho Joint Photo-graphic Experts Groups) là một trong những phương pháp nén ảnh hiệu quả, có tỷ lệ nén ảnh tới vài chục lần. Tuy nhiên ảnh sau khi giải nén sẽ khác với ảnh ban đầu. Chất lượng ảnh bị suy giảm sau khi giải nén. Sự suy giảm này tăng dần theo hệ số nén. Tuy nhiên sự mất mát thông tin này có thể chấp nhận được và việc loại bỏ những thông tin không cần thiết được dựa trên những nghiên cứu về hệ nhãn thị của mắt người. Phần mở rộng của các file JPEG thường có dạng .jpeg, .jfif, .jpg, .JPG hay .JPE; dạng .jpg là dạng được dùng phổ biến nhất. Hiện nay dạng nén ảnh JPEG rất được phổ biến trong ĐTDD cũng như những trang thiết bị lưu giữ có dung lượng nhỏ. Công đoạn chính là chia nhỏ bức ảnh thành nhiều vùng nhỏ (thông thường là những vùng 8x8 pixel) rồi sử dụng biến đổi cosin rời rạc để biến đổi những vùng thể hiện này thành dạng ma trận có 64 hệ số thể hiện "thực trạng " các pixel. Điều quan trọng là ở đây hệ số đầu tiên có khả năng thể hiện "thực trạng " cao nhất , khả năng đó giảm rất nhanh với các hệ số khác. Nói cách khác thì lượng thông tin của 64 pixels tập trung chủ yếu ở một số hệ số ma trận theo biến đổi trên. Trong giai đoạn này có sự mất mát thông tin, bởi không có biến đổi ngược chính xác. Nhưng lượng thông tin bị mất này chưa đáng kể so với giai đoạn tiếp theo. Ma trận nhận được sau biến đổi cosin rời rạc được lược bớt sự khác nhau giữa các hệ số. Đây chính là lúc mất nhiều thông tin vì người ta sex vứt bỏ những thay đổi nhỏ của các hệ số. Như thế khi bung ảnh đã nén ta sẽ có được những tham số khác của các pixel. Các biến đổi trên áp dụng cho thành phần U và V của ảnh với mức độ cao hơn so với Y (mất nhiều thông tin của U và V hơn). Sau đó thì áp dụng phương pháp mã hóa của Gernot Hoffman: phân tích dãy số,

các phần tử lặp lại nhiều được mã hóa bằng các ký hiệu ngắn (marker). Khi bung ảnh người ta chỉ việc làm lại các bước trên theo quá trình ngược lại cùng với các biến đổi ngược

Hình 3.5: Ảnh dạng JPEG

2. Các kiểu hình ảnh trong Matlab

Image Processing Toolbox của Matlab hỗ trợ bốn kiểu biểu diễn hình ảnh cơ bản gồm: Ảnh chỉ số (index images), ảnh độ sáng (intensity images), ảnh nhị phân (binary images), ảnh RGB (RGB images).

Ảnh chỉ số:

Với cách biểu diễn ảnh này, mỗi ảnh sẽ được biểu diễn bởi hai ma trận, một ma trận dữ liệu ảnh X và một ma trận màu (còn gọi là bản đồ màu). Ma trận dữ liệu có thể thuộc kiểu uint8, uint16, hoặc double. Ma trận màu là ma trận kích thước mx3 gồm các phần tử kiểu double có giá trị nằm trong khoảng [0, 1]. Mỗi hàng của ma trận xác định các thành phần red, green, blue của một màu trong tổng số m màu được sử dụng trong ảnh, giá trị của mỗi phần tử trong ma trận dữ liệu cho biết màu của điểm ảnh đó nằm ở hàng nào trong ma trận màu. Nếu ma trận dữ liệu thuộc về kiểu double, giá trị 1 sẽ tương ứng với hạng thứ 1 trong bằng màu, giá trị thứ 2 sẽ tương ứng với hàng thứ 2 trong bảng màu. Nếu ma trận dữ liệu thuộc kiểu uint8 hoặc uint16 thì giá trị 0 tương ứng với hàng 1, giá trị 1 tương ứng với hàng 2... Riêng với kiểu uint6, Matlab không

hỗ trợ đủ các phép toán so với kiểu uint8 nên khi cần xử lý ta chuyển sang kiểu dữ liệu uint8 hoặc double bằng các hàm imapprox hoặc im2double.

Ảnh biểu diễn theo độ sáng:

Mỗi ảnh được biểu diễn bởi một ma trận hai chiều, trong đó giá trị của mỗi phần tử cho biết độ sáng (hay mức xám) của điểm ảnh đó. Ma trận này có thể thuộc một trong các kiểu uint8, uint16 hoặc double. Trong đó giá trị nhỏ nhất 0 tương ứng với màu đen còn giá trị lớn nhất (255 hoặc 655535 tùy kiểu dữ liệu nào) ứng với màu trắng. Như vậy, ảnh biểu diễn theo kiểu này gọi là "ảnh trắng đen" hoặc ảnh gray scale.

Ảnh nhị phân:

Ảnh nhị phân cũng được biểu diễn bằng ma trận hai chiều nhưng thuộc kiểu logical, có nghĩa là mỗi điểm ảnh chỉ có thể nhận một trong hai giá trị 0 (đen) hoặc 1 (trắng).

Ảnh RGB:

Ảnh RGB còn gọi là ảnh "true color" do tính trung thực của nó. Ảnh này được biểu diễn bởi một ma trận 3 chiều có kích thước $m \times n \times 3$, với $m \times n$ là kích thước ảnh theo pixels. Ma trận này định nghĩa các thành phần màu red, green, blue cho mỗi điểm ảnh, các phần tử của nó có thể thuộc kiểu uint8, uint16, hoặc double. Ví dụ, điểm ảnh ở vị trí (10, 5) sẽ có ba thành phần màu được xác định bởi các giá trị (10, 5, 1), (10, 5, 2) và (10, 5, 3). Các file ảnh hiện nay thường sử dụng 8 bit cho thành phần màu, nghĩa là mất 24 bit cho mỗi điểm ảnh (khoảng 16 triệu màu).

3. Chuyển đổi giữa các kiểu dữ liệu

Chúng ta có thể chuyển đổi giữa các kiểu dữ liệu uint8, uint16 và double nhờ sử dụng các hàm chuyển đổi của Matlab như im2double, im2uint8, im2uint16. Cú pháp của các hàm này rất đơn giản, chỉ cần nhập vào ma trận cần chuyển kiểu, riêng với ảnh indexed cần thêm vào chuỗi "indexed".

Tuy nhiên cần lưu ý các vấn đề sau khi chuyển đổi ảnh:

- Khi chuyển đổi từ ảnh nhiều bit sang ảnh ít bit hơn, như chuyển từ uint16 sang uint8 thì sẽ mà mất đi một số thông tin của ảnh ban đầu, chất lượng ảnh sẽ giảm.
- Khi chuyển đổi dữ liệu với kiểu indexed, thì lưu ý các thông tin ma trận là địa chỉ trong bảng đổ màu chứ không phải giá trị màu nên không phải lúc nào cũng chuyển đổi được. Muốn chuyển đổi được đầu tiên ta phải dùng hàm

imapprox để giảm số màu cần biểu diễn ảnh xuống (bằng cách cho các màu gần giống nhau thành một) rồi mới chuyển.

Tên thuộc tính	Mô tả
Filename	Chuỗi chứa tên file
FileModDate	Ngày chỉnh file gần nhất
FileSize	Số nguyên chỉ kích thước(byte)
Format	Chuỗi cho biết định dạng ảnh
FormatVersion	Tên phiên bản định dạng ảnh
Width	Chiều rộng ảnh (pixel)
Height	Chiều cao ảnh (pixel)
BitDepth	Số bit trên một pixel
ColorType	Cho biết kiểu ảnh (truecolor, indexed...)

Bảng 3.1 Các thông tin khi gọi hàm iminfo

4. Các phép toán số học cơ bản đối với dữ liệu ảnh

Các phép toán bao gồm các phép cộng, trừ, nhân và chia. Đây là các thao tác xử lý ảnh cơ bản trước khi thực hiện các phép biến đổi phức tạp khác. Người sử dụng có thể sử dụng các hàm số học mà Matlab cung cấp để tác động lên dữ liệu ảnh. Tuy nhiên Matlab chỉ hỗ trợ các phép toán này trên kiểu dữ liệu double nên cần phải chuyển đổi trước khi thực hiện. Để đơn giản hơn, Matlab cung cấp các hàm thực hiện các phép toán số học có thể chấp nhận bất kỳ kiểu dữ liệu ảnh nào và trả về kết quả giá trị thuộc cùng kiểu với các toán hạng.

Cú pháp	Mô tả
Z = imabsdiff(x,y)	Trừ tương ứng mỗi phần tử y cho mỗi phần tử của x, trả về trị tuyệt đối hiệu

Z = imadd(x,y,out_class)	Cộng hai ảnh, cộng ảnh với hằng số, out_class kiểu dữ liệu tổng.
im2 = imcoplement(im)	Lấy bù của ảnh im
Z = imdivide(x,y)	Chia các phần tử x cho các phần tử y, kết quả làm tròn
Z = imlincomb(i1,k1,k2,a2,..,out_class)	Lấy tổ hợp tuyến tính $z = k1 * a1 + k2 * a2 + \ldots$
Z = immultiply(x,y)	Nhân hai ảnh, ảnh với hằng số
Z = imsubtrace(x,y)	Trừ hai ảnh, ảnh với hằng số

Bảng 3.2 Các phép toán số học trên ảnh

5. Các hàm hiển thị ảnh trong Matlab

Để hiển thị ảnh, Matlab cung cấp 2 hàm cơ bản là image và imagesc. Ngoài ra trong Image Processing Toolbox cũng có hai hàm hiển thị khác là imview và imshow.

- Hàm image(x, y, c) để hiển thị hình ảnh biểu diễn bởi ma trận c kích thước mxn lên hệ trục tọa độ. x, y là các vector xác định vị trí của các điểm c(1, 1) và c(m, n).
- Hàm imagesc có chức năng tương tự hàm image, ngoại trừ việc dữ liệu ảnh sẽ được co giãn để sử dụng toàn bộ bản đồ màu hiện hành.
- Hàm imview cho phép hiển thị ảnh trên cửa sổ riêng nền Java, gọi là Image Viewer.
- Hàm imshow cho phép hiển thị ảnh trên một Figure và tự động thiết lập giá trị các đối tượng image, axes, figure để hiển thị hình ảnh.

Các hàm chuyển đổi loại ảnh và kiểu dữ liệu ảnh	
dither	Tạo ảnh nhị phân hay RGB
gray2ind	Chuyển ảnh đen trắng thành ảnh indexed
grayslice	Chuyển ảnh trắng đen thành ảnh indexed bằng lấy ngưỡng
im2bw	Chuyển ảnh thành ảnh kiểu dữ liệu nhị phân

im2double	Chuyển ảnh thành ảnh kiểu dữ liệu double
im2uint16	Chuyển ảnh thành ảnh kiểu dữ liệu uint16
im2uint8	Chuyển ảnh thành ảnh kiểu dữ liệu uint8
imapprox	Xấp xỉ ảnh indexed bằng cách giảm số màu
ind2gray	Chuyển ảnh indexed thành ảnh gray scale
ind2rgb	Chuyển ảnh indexed thành ảnh RBG
mat2gray	Tạo ảnh gray scale từ ma trận
rgb2ind	Chuyển ảnh RBG thành ảnh indexed
grb2gray	Chuyển ảnh RBG thành ảnh gray scale
Các hàm truy xuất dữ liệu ảnh	
imfinfo	Truy xuất thông tin ảnh
imread	Đọc ảnh từ file và xuất ra ma trận ảnh
imwrite	Lưu ma trận ảnh thành file ảnh
Các hàm biến đổi hình học	
cp2tform	Định nghĩa phép biến đổi hình học từng cặp tương ứng
imcrop	Trích xuất một phần ảnh
imresize	Thay đổi kích thước ảnh
imrotate	Thực hiện phép quay ảnh
imtransform	Thực hiện phép biến đổi hình học tổng quát
maketform	Định nghĩa phép biến đổi hình học tổng quát

Bảng 3.3 Các hàm xử lý hình ảnh khác trong Matlab

6. Các hàm khác được sử dụng trong đề tài

- *T = ls(pathfolder):* Lấy tất cả tên file trong đường dẫn thư mục pathfolder
- *D = size(a):* Trả về giá trị là ma trận có dạng [x,y] là kích thước của ma trận a
- *T = reshape(X, M, N):* Trả về ma trận có kích thước MxN, với các phần tử là các phần tử nằm trong ma trận X.
- *mean(X, dim):* với dim là chiều lấy trung bình, nếu dim bằng 1 lấy trung bình theo cột, nếu dim bằng 2 lấy trung bình theo hàng. Không có tham số thì dim mặc định bằng 1.
- *double(X):* Ép kiểu X sang kiểu *double*.
- *[V, D] = eig(X):* Tạo ra một ma trận đường chéo D của các giá trị riêng và một ma trận V có các cột tương ứng là các vector riêng, do đó: $X * V = V * D$.
- *Diag(V):* Lấy các phần tử trong đường chéo chính, kết quả trả về là một mảng các phần tử trên đường chéo chính của V.
- *Sort(X):* sắp tăng dần hay giảm. Đối với vector, Sort(X) sắp xếp các phần tử của X thứ tự tăng dần. Đối với ma trận, Sort(X) xếp loại mỗi cột của X theo thứ tự tăng dần. Khi X là một mảng di động của chuỗi, Sort(X) sắp xếp các ký tự theo thứ tự bảng mã ASCII.
- *Orth(X):* Xây dựng một hệ cơ sở trực chuẩn.
- *Norm(A-B):* Hàm tính khoảng cách ơ clit giữa hai vector A và B.

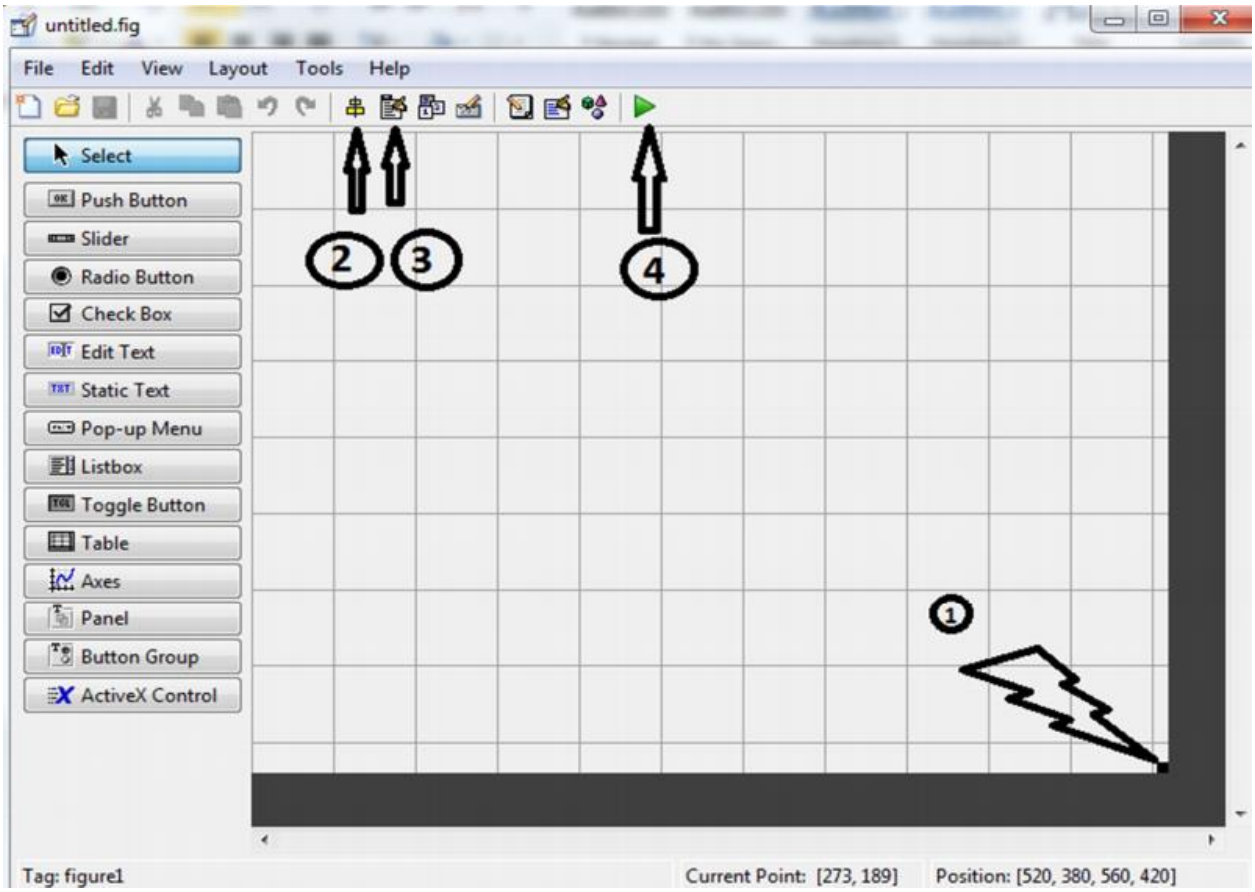
II. GIAO DIỆN GUI CỦA MATLAB

1. Giao diện GUI

GUI (Graphical User Interface) là giao diện đồ họa có điều khiển bởi nhiều thanh công cụ được người lập trình tạo sẵn, cho tương tác giữa người dùng là giao diện chương trình, Mỗi chương trình được người lập trình tạp sẵn giao diện thực hiện một vài chức năng được người lập tình tạo sẵn và giao tiếp với người sử dụng.

- *Khởi động:* Trong cửa sổ Comment Windowns gõ lệnh "guide" và enter ta được giao diện màn hình cơ bản của GUI:
- *Chọn Dòng "Blank GUI (Default)"* để tạo một giao diện gui bắt đầu với giao diện trống. Các dòng còn lại để khởi động GUI với một giao diện được tạo sẵn. Nhấp chọn OK để tạo một giao diện bắt đầu với giao diện trống. Ta được hình ảnh giao diện trong GUI.

- *Trước khi tạo giao diện ta lưu File lại*, Matlab sẽ tự động lưu 2 file, một file đuôi .m và một file đuôi .fig. hoặc ta có thể nhấn F5, Matlab sẽ chuyển đường dẫn đến thư mục lưu file, chọn nơi cần lưu và nhấn Save.



Các thao tác cơ bản:

1. Kéo thả để thay đổi độ rộng của giao diện.
2. Căn chỉnh các nút, biểu tượng trên giao diện.
3. Tạo giao diện con liên kết với giao diện chính.
4. Nút Run để thực thi chương trình.

Các conltrol được Matlab GUI hỗ trợ sẵn (bên phải hình):

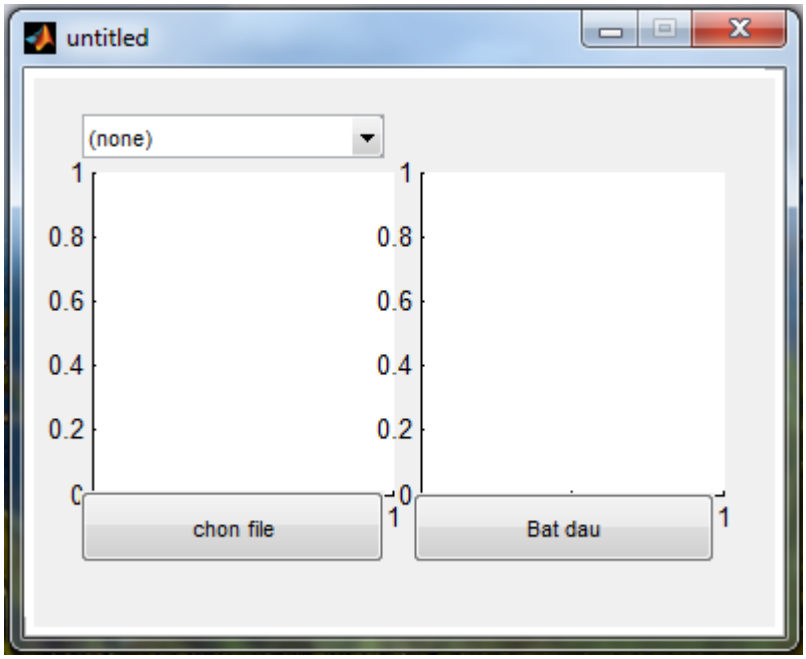
- *Push Button:* là nút nhấn, khi nhấn vào sẽ thực thi lệnh trong cấu trúc hàm callback của nó
- *Slider:* là thanh trượt cho phép người dùng di chuyển thanh trượt để thục thi lệnh.

- *Radio Button:* Nó giống như Check Box nhưng thường được sử dụng để tạo sự lựa chọn duy nhất, tức là 1 lần chỉ được chọn 1 trong số các nhóm nhiều nút. Khi một ô được chọn thì các ô còn lại trong nhóm bị bỏ chọn.
- *Check box:* Sử dụng để đánh dấu tích (thực thi) vào và có thể check nhiều ô để thực thi
- *Edit Text:* là nơi các kí tự được nhập vào từ người dùng, người dùng có thể thay đổi được
- *Static Text:* Là các kí tự được hiển thị thông qua các callback, hoặc thông thường để viết nhãn cho các biểu tượng, người dùng không thể thay đổi nội dung.
- *Pop-up Menu:* mở ra danh sách các lực chọn khi người dùng nhấp chuột vào. Chỉ chọn được 1 mục trong danh sách các mục.
- *List Box:* hộp thoại danh sách cách mục, cho phép người dùng chọn một hoặc nhiều mục.
- *Toggle Button:* là nút nhấn có 2 điều khiển, khi nhấp chuột và nhả ra, nút nhấn được giữ và lệnh thực thi, khi nhấp chuột vào lần thứ 2, nút nhấn nhả ra, hủy bỏ lệnh vừa thực thi.
- *Table:* tạo ra một bảng tương tự trong Excel.
- *Axes:* Đây là giao diện đồ họa hiển thị hình ảnh, nó có nhiều thuộc tính bao gồm: không gian 2D (theo trục đứng và trục ngang), 3D (hiển thị không gian 3 chiều)
- *Panel:* Tạo ra một mảng nhóm các biểu tượng lại với nhau giúp ta dễ kiểm soát và thao tác khi di chuyển
- *Button Group:* quản lí sự lựa chọn của nút Radio Button.
- *Active Control:* Quản lí một nhóm các bút hoặc các chương trình liên quan với nhau trong Active.

Ta double click vào bất kì một biểu tượng đã tạo để điều chỉnh thuộc tính bên trong. Một số thuộc tính quan trọng:

- *Color:* điều chỉnh màu sắc.
- *String:* Hiển thi chữ trên nền của biểu tượng.
- *Style:* thay đổi thuộc tính biểu tượng.
- *Tag:* Khi tác động vào sẽ gọi hàm mà có tag ở trong code chương trình.

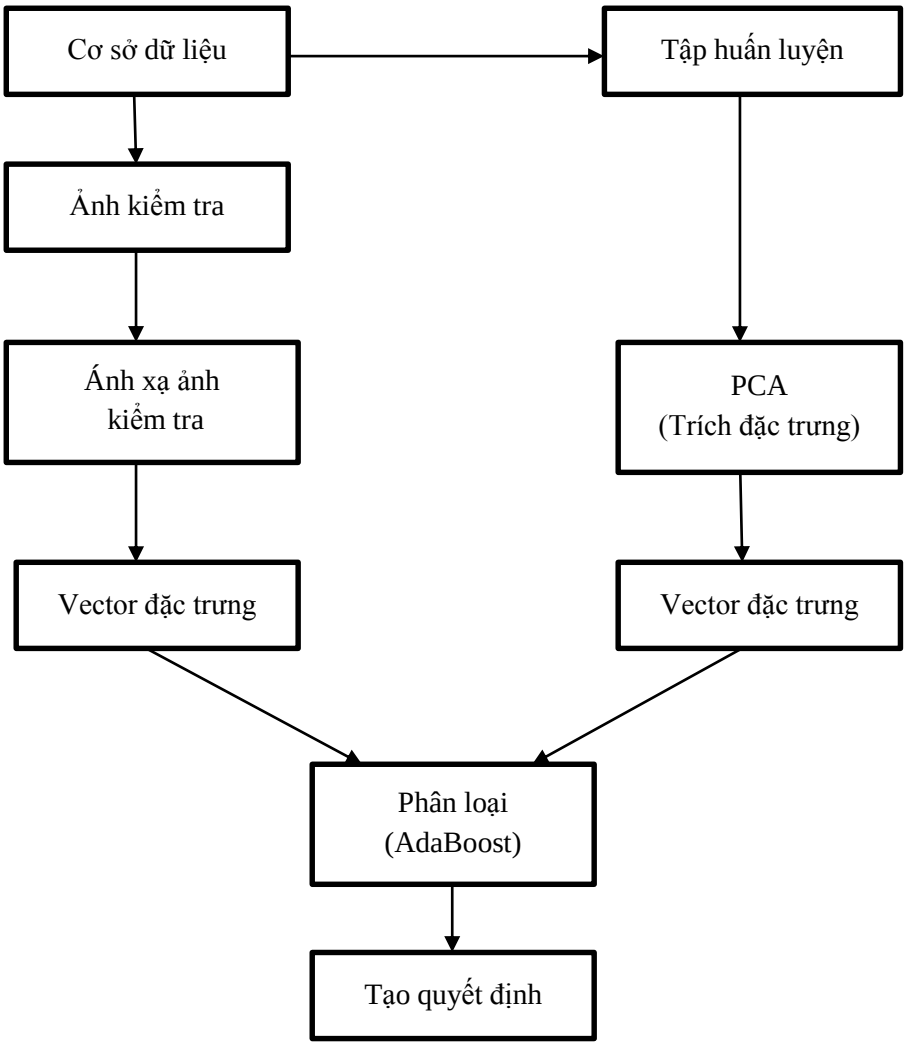
2. Giao diện chương trình nhận dạng mặt người



Thực hiện chạy guide đã thiết kế, giao diện như hình trên. Giao diện guide gồm 3 control chính là button, Axes và Pop_up menu. Sau khi bấm nut chọn file và ấn vào mũi tên xổ xuống trong pop_up menu ta sẽ có một loạt các đường dẫn file ảnh *.jpg hiện có trong thư muc E:\image2. Khi lựa chọn một đường dẫn trong pop_up ảnh tương ứng sẽ hiện ra trên Axes1, ảnh này chính là ảnh đầu vào của quá trình nhận diện khuôn mặt. Sau khi đã hiện ảnh đầu vào, ta bấm nút Bat dau, thuật toán PCA được thực hiện và tìm ra ảnh giống nhất trong cơ sở dữ liệu, nếu không tìm thấy ảnh, ảnh hiện ra sẽ có màu đen.

III. SƠ ĐỒ KHỐI VÀ CODE CHƯƠNG TRÌNH

1. Sơ đồ khối



Sơ đồ khối tổng quát của chương trình

2. Code chương trình.

Ai muốn mua code liên hệ sđt: 0964435482