Đề bài:

Tìm hiểu các đặc trưng sinh trắc học khuôn mặt, nghiên cứu ứng dụng của phép biến đổi KL và phân tích thành các thành phần chính (PCA) trong trích chọn đặc trưng khuôn mặt.

Khảo sát một ứng dụng nhận dạng khuôn mặt bằng phương pháp PCA

Vai trò của KL, PCA:

Đóng vai trò xác định tổ hợp các véc tơ riêng, trị riêng (khuôn mặt riêng). 🡪 Mỗi khuôn mặt là tổ hợp tuyến tính của các khuôn mặt riêng.

1. Mục tiêu

* Khảo sát một ứng dụng ứng dụng nhận dang khuôn mặt theo phương pháp PCA để thấy được ứng dụng của phép biến đổi KL, PCA trong trích chọn đặc trưng khuôn mặt.
* Mục tiêu của ứng dụng: nhận dạng chính xác nhất khuôn mặt thử nghiệm dựa trên những khuôn mặt đã có sẵn

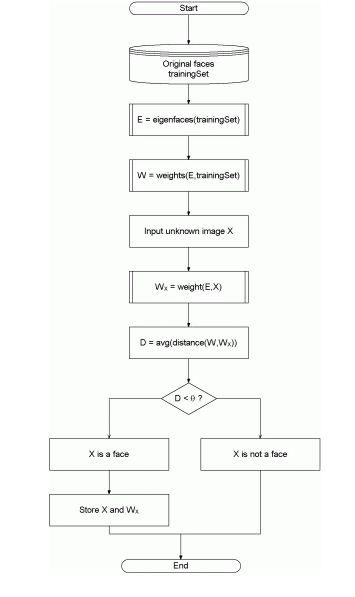
1. Môi trường

Matlab

1. Phân tích

Ý tưởng của phương pháp là phân tích các thành phần chủ yếu tạo nên khuôn mặt, hoặc những vectơ riêng của ma trận hiệp phương sai của tập ảnh các khuôn mặt . Mỗi vectơ chưa một số lượng các đặc tính riêng biệt của khuôn mặt từ đó tạo ra sự khác nhau giữa các khuôn mặt. Bên cạnh đó các vectơ riêng này có thể được biểu diễn giống với một khuôn mặt, bởi vậy, các vectơ này còn được gọi là các khuôn mặt riêng – eigenfaces.

Các bước thực hiện của giải thuật:



* Yêu cầu tạo khởi tạo một tập dữ liệu bao gồm các khuôn mặt (tập dữ liệu huấn luyện)
* Tính các khuôn mặt riêng từ tập dữ liệu huấn luyện đã có nhưng chỉ giữ lại M ảnh tương ứng với M trị riêng lớn nhất. Các vectơ riêng tương ứng với các trị riêng này tạo nên không gian mặt.
* Tính toán trọng số không gian của các nhóm khuôn mặt tương ứng trong cơ sở dữ liệu bằng cách chiếu lên không gian mặt.
* Tính toán tập trong số của khuôn mặt cần nhận dạng bằng cách chiếu lên M khuôn mặt riêng đã có.
* Quyết định xem hình ảnh đưa vào có phải là ảnh khuôn mặt hay không.
* Nếu ảnh đưa vào là ảnh khuôn mặt, dựa vào trọng số đã tính ở trên, xác định xem ảnh có thuộc các nhóm khuôn mặt đã biết hay không.
* Nếu ảnh khuôn mặt này tương ứng với nhóm khuôn mặt đã có trong cơ sở dữ liệu, tính toán cập nhật các khuôn mặt riêng và các trọng số riêng (bước này có thể bỏ qua trong một số ứng dụng)
* Nếu một khuôn mặt lạ xuất hiện nhiều lần, ta có thể tập hợp lại thành nhóm và tạo mới trong cơ sở dữ liệu.

Như vậy, trong ứng dụng, KL, PCA đóng vai trò xác định tổ hợp các vectơ riêng – khuôn mặt riêng.

Chi tiết các bước thực hiện:

1. Tính các vectơ khuôn mặt riêng:

Giả sử ta có ảnh I(x,y) là một mảng 2 chiều NxN các giá trị cường độ (8bits), có thể coi ảnh như một vector 1 chiều N2, do đó với một ảnh có kích cỡ điển hình 256x256, vector một chiều của ảnh là 256x256 = 65 536, hay tương ứng với một điểm trong không gian chiều 65 536. Do đó, một tập các ảnh được tham chiếu sang một tập các điểm trong không gian này.

Để có được các chi tiết cần thiết, ảnh khuôn mặt cần được biểu diễn trong một không gian nhỏ hơn. Tương đương với việc ta cần tính các vector riêng của ma trận hiệp phương sai (các khuôn mặt riêng).

Gọi tập ảnh ban đầu là Γ1, Γ2, Γ3,…, ΓM. Khuôn mặt trung bình của tập được ký hiệu là

http://www.pages.drexel.edu/~sis26/Eigenface%20Tutorial_files/image004.gif

Mỗi khuôn mặt chênh lệch với khuôn mặt trung bình một đại lượng là vector :

http://www.pages.drexel.edu/~sis26/Eigenface%20Tutorial_files/image006.gif

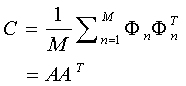
Tập các vector này được phân tích để xác định M vector trực giao un có thể biểu diễn tốt nhất tập dữ liệu. Vector thứ k, uk, được lựa chọn sao cho:

http://www.pages.drexel.edu/~sis26/Eigenface%20Tutorial_files/image008.gif

có giá trị cao nhất, với ràng buộc :

http://www.pages.drexel.edu/~sis26/Eigenface%20Tutorial_files/image010.gif

Vector uk và λk tương ứng là vector riêng và trị riêng của ma trận hiệp phương sai:



http://www.pages.drexel.edu/~sis26/Eigenface%20Tutorial_files/image014.gif

Như vậy, ma trận C có kích thước N­2x N­2 , việc xác định N­2 vector riêng và giá trị riêng là khó thực hiện với kích thước ảnh điển hình. Ta cần một phương pháp dễ dàng hơn để tìm các vector riêng này. Ta chỉ giữ lại những vectơ riêng tương ứng với những trị riêng lớn nhất bởi những vectơ riêng này chứa nhiều đặc trưng nhất, còn các vectơ ứng với các trị riêng nhỏ chỉ chứa một số ít các đặc trưng.

Nếu số điểm dữ liệu trong không gian ảnh nhỏ hơn chiều không gian (M<N2), sẽ chỉ có M – 1 vector riêng có ý nghĩa. Ta sẽ tính các vector riêng N2 chiều bằng cách tính các vector riêng của ma trận MxM. Xét các vector riêng vi của ATA :

ATAvi = μi vi

Nhân 2 vế với A , ta có:

AAT Avi = μi Avi

Từ đó ta thấy Avi là vector riêng của C = AAT.

Tiếp theo, ta thiết lập ma trận MxM : L=ATA, với Lmn=Φn, và tìm M vector riêng, vl, của L. Các vector này xác định tổ hợp tuyến tính M ảnh từ tập ban đầu để tạo ra các mặt riêng ul:

http://www.pages.drexel.edu/~sis26/Eigenface%20Tutorial_files/image018.gif

Bằng phương pháp này, các phép tính đã được giảm đi đáng kể.

Thực tế, tập ảnh khuôn mặt huấn luyện là tương đối nhỏ (M«N2), vì vậy các phép toán trở nên dễ dàng hơn. Các trị riêng tương ứng cho phép ta xếp loại các vector riêng theo mức độ hữu ích của chúng trong biểu thị các điểm khác biệt giữa các ảnh.



*Tập khuôn mặt đào tạo (training set).*



*Khuôn mặt trung bình Ψ Các khuôn mặt riêng được lựa chọn*

1. Sử dụng các vectơ khuôn mặt riêng để phân loại một khuôn mặt

Do các khuôn mặt riêng tương ứng với các trị riêng lớn nhất được chọn là khá đầy đủ để mô tả ảnh các khuôn mặt, nên ta có thể sử dụng chúng như một công cụ nhận dạng khuôn mặt. Thực tế, một số lượng M’ nhỏ hơn tỏ ra hiệu quả trong nhận dạng, do ta không cần thiết phải tái tạo lại ảnh ban đầu. Các khuôn mặt riêng tạo ra một không gian chiều con M’ từ không gian ảnh N2 ban đầu. Các vector riêng quan trọng M’ của ma trận L được lựa chọn từ những vector có trị riêng lớn nhất. Trong nhiều thực nghiệm, với cơ sở M=16 ảnh khuôn mặt, M’=7 khuôn mặt riêng được sử dụng.

Một ảnh mặt mới (Γ) được biến đổi thành các thành phần khuôn mặt riêng (chiếu vào không gian ảnh mặt) bằng một biến đổi đơn giản

http://www.pages.drexel.edu/~sis26/Eigenface%20Tutorial_files/image020.gif

Với k = 1,…, M’. uT­k là các vectơ riêng ta đã tính ở trên

http://www.pages.drexel.edu/~sis26/Eigenface%20Tutorial_files/image022.gif

Các trọng số của một vector ΩT = [ω1, ω2, …, ωM’ ] mô tả các phần liên quan của mỗi khuôn mặt riêng trong biểu diễn ảnh khuôn mặt nhập vào; coi các khuôn mặt riêng như môt tập cơ sở các ảnh khuôn mặt. Vector này có thể được sử dụng trong giải thuật nhận dạng chi tiết chuẩn để tìm ra một số các lớp khuôn mặt định trước. Phương pháp đơn giản nhất để xác định lớp khuôn mặt nào mô tả tốt nhất ảnh khuôn mặt nhập vào là tìm lớp khuôn mặt k làm cực tiểu hóa khoảng cách Euclide:

http://www.pages.drexel.edu/~sis26/Eigenface%20Tutorial_files/image024.gif

Ωk là một vector mô tả lớp khuôn mặt thứ k. Các lớp khuôn mặt Ωi được tính bằng cách lấy trung bình các kết quả của biểu diễn khuôn mặt riêng trên một số lượng nhỏ ảnh các khuôn mặt của mỗi cá nhân. Một khuôn mặt được coi là thuộc vào lớp k nếu giá trị εk cực tiểu nhỏ hơn một ngưỡng θk chọn trước. Nếu không, khuôn mặt sẽ được xếp vào loại “chưa biết”, và có thể được sử dụng để tạo ra một lớp khuôn mặt mới.

Từ đây, ta có 3 khả năng cho hình ảnh đầu vào và vector mẫu của ảnh đó:

1. Gần không gian mặt và gần một lớp khuôn mặt. 🡪 Một cá nhân được nhận dạng và xác định.
2. Gần không gian mặt như không gần một lớp khuôn mặt. 🡪 Phát hiện một cá nhân lạ.
3. Xa không gian mặt và gần một lớp mặt. 🡪 Ảnh đã cho không diễn tả gương mặt.
4. Xa không gian mặt và không gần lớp mặt nào. 🡪 Ảnh đã cho không diễn tả khuôn mặt.
5. Triển khai và kết quả thực hiện

Tập các ảnh đầu vào:

* 30 bức ảnh thể hiện các trạng thái biểu cảm khác nhau của 3 người.
* Mỗi bức ảnh có kích cỡ là: 112x92 (10304 pixels)







Đọc dữ liệu từ tập ảnh đầu vào:

k = 0;

for i=1:1:3

for j=1:1:10

filename = sprintf('C:\\Users\\PHUC\\Desktop\\KL\_based Face Recognition System\\att\_faces\\s%d\\%d.pgm',i,j); %traning database path

image\_data = imread(filename);

k = k + 1;

x(:,k) = image\_data(:);

anot\_name(k,:) = sprintf('%2d:%2d',i,j); % for plot annotations

end;

end;

nImages = k; %total images

imsize = size(image\_data); %image size

nPixels = imsize(1)\*imsize(2); %total pixels per image

x = double(x)/255; %convert tu double sang thuong

Tính toán các vectơ khuôn mặt riêng (eigenvalue)

* Tính khuôn mặt trung bình

%Calculate mean\_face - average\_face

avrgx = mean(x')';

for i=1:1:nImages

x(:,i) = x(:,i) - avrgx; % substruct the average

end;

subplot(2,2,1); imshow(reshape(avrgx, imsize)); title('mean face')



*Ảnh khuôn mặt trung bình tính từ tập ảnh huấn luyện trên*

* Tính ma trận hiệp phương sai và các vectơ riêng (vectơ khuôn mặt riêng)

%Calculate covariance matrix

cov\_mat = x'\*x;

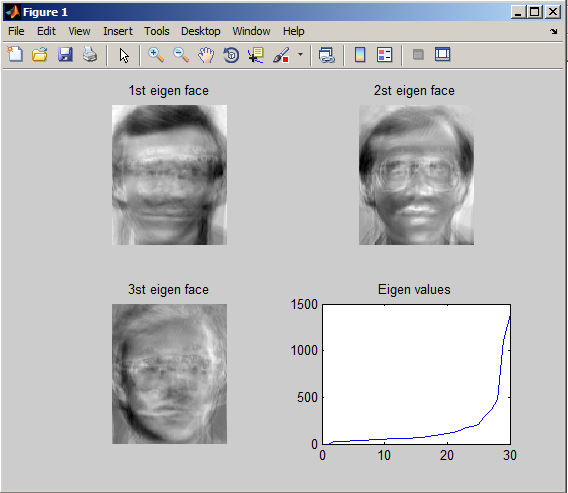
[V,D] = eig(cov\_mat); %eigenvalues

V = x\*V\*(abs(D))^-0.5;

subplot(2,2,2); imshow(ScaleImage(reshape(V(:,nImages ),imsize))); title('1st eigen face');

subplot(2,2,3); imshow(ScaleImage(reshape(V(:,nImages-1),imsize))); title('2st eigen face');

subplot(2,2,4); plot(diag(D)); title('Eigen values');



*ví dụ về 3 vectơ khuôn mặt riêng (trên tổng số 30 vectơ khuôn mặt riêng)*

* Tiến hành nhận dạng với một ảnh, ảnh được chọn thử nghiệm có chỉ số image\_index bằng 5.

image\_index = 5; %index of representation image

reconst = V\*KLCoef';

diff = abs(reconst(:,image\_index) - x(:,image\_index));

strdiff\_sum = sprintf('delta per pixel: %e',sum(sum(diff))/nPixels);

figure;

subplot(2,2,1); imshow((reshape(avrgx+reconst(:,image\_index), imsize))); title('Reconstructed');

subplot(2,2,2); imshow((reshape(avrgx+x(:,image\_index), imsize)));title('original');

subplot(2,2,3); imshow(ScaleImage(reshape(diff, imsize))); title(strdiff\_sum);

for i=1:1:nImages

dist(i) = sqrt(dot(KLCoef(image\_index,:)-KLCoef(i,:), KLCoef(image\_index,:)-KLCoef(i,:))); %euclidean

end;

subplot(2,2,4); plot(dist,'.-'); title('euclidean distance from the first face');

% MODE: face:expression

%VD : s1/2.pgm

figure;

show\_faces = 1:1:nImages/2;

plot(KLCoef(show\_faces,nImages), KLCoef(show\_faces,nImages-1),'.'); title('Desomposition: Numbers indicate (Face:Expression)');

for i=show\_faces

name = anot\_name(i,:);

text(KLCoef(i,nImages), KLCoef(i,nImages-1),name,'FontSize',8);

end;

%image\_index = 12;

for i=1:1:nImages

dist\_comp(i) = sqrt(dot(KLCoef(image\_index,:)-KLCoef(i,:), KLCoef(image\_index,:)-KLCoef(i,:))); %euclidean

strDist(i) = cellstr(sprintf('%2.2f\n',dist\_comp(i)));

end;

[sorted, sorted\_index] = sort(dist\_comp); % sort distances

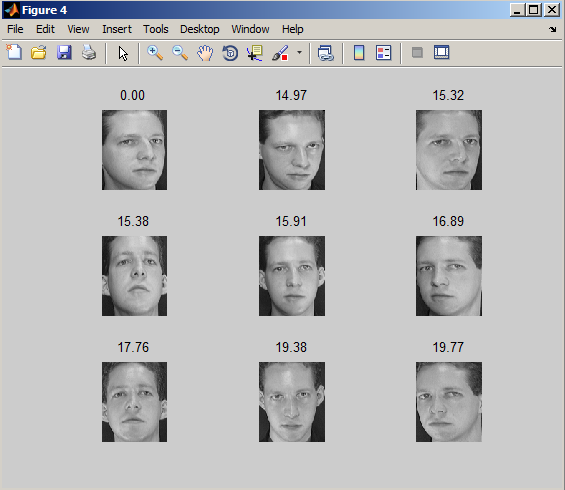
figure; % open new figure

for i=1:1:9

subplot(3,3,i); imshow((reshape(avrgx+x(:,sorted\_index(i)), imsize))); title(strDist(sorted\_index(i)));

end;

Sau khi tính toán các khoảng cách Euclide của khuôn mặt cần nhận dạng với các khuôn mặt khác ta thu được kết quả:

**

*Các khuôn mặt có khoảng các Euclide gần nhất với khuôn mặt cần nhận diện*

Khuôn mặt có khoảng cách Euclide bằng 0 chính là khuôn mặt cần nhận dạng, trong thực tế, không bao giờ khuôn mặt cần nhận dạng cũng có trong tập thử nghiệm, vì vậy, để nhận dạng một ảnh khuôn mặt, người ta sẽ chọn khuôn mặt trong tập thử nghiệm có khoảng cách Euclide ngắn nhất (nhỏ nhất) với khuôn mặt cần nhận dạng.