```
%This code to apply PCA (Principal Component Analysis)

% Remember that each column of the data matrix(input matrix) represent one image or
pattern
% Note: the data here represent two classes
% Class 1: data(:,1:4)
% Class 2: data(:,5:8)

data = [1    1    2    0    7    6    7    8
      3    2    3    3    4    5    5    4];

[r,c] = size(data);

% Compute the mean of the data matrix "The mean of each row"
m = mean(data')';

% Subtract the mean from each image [Centering the data]
d=data-repmat(m,1,c);


% Compute the covariance matrix (co)
co=d*d';

% Compute the eigen values and eigen vectors of the covariance matrix
[eigvector,eigvl]=eig(co);


% Sort the eigen vectors according to the eigen values
eigvalue = diag(eigvl);
[junk, index] = sort(eigvalue,'descend');
eigvalue = eigvalue(index);
eigvector = eigvector(:, index);

% Compute the number of eigen values that greater than zero (you can select any
threshold)
count1=0;
for i=1:size(eigvalue,1)
   if(eigvalue(i)>0)
      count1=count1+1;
   end
end


% And also we can use the eigen vectors that the corresponding eigen values is greater than
zero(Threshold) and this method will decrease the
% computation time and complexity
```

```
vec=eigvector(:,1:count1);

% Compute the feature matrix (the space that will use it to project the testing image on it)
x=vec'*d;


% If you have test data do the following
t=[1;1]  % this test data is close to the first class

%Subtract the mean from the test data
t=t-m;

%Project the testing data on the space of the training data
t=vec'*t;

% Then if you want to know what is the class of this test data?  just use
% any classifier (In our case we used minimum distance classifier)

alldata=t';
alldata(2:size(x,2)+1,:)=x'
dist=pdist(alldata);
[a,b]=min(dist(:,1:size(x,2)))
```