1 Principal component analysis (PCA)

Principal component analysis (PCA) is a mathematical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has as high a variance as possible (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it be orthogonal to (uncorrelated with) the preceding components

PCA is the simplest of the true eigenvector-based multivariate analyses. Often, its operation can be thought of as revealing the internal structure of the data in a way which best explains the variance in the data. If a multivariate dataset is visualised as a set of coordinates in a high-dimensional data space (1 axis per variable), PCA can supply the user with a lower-dimensional picture, a "shadow" of this object when viewed from its (in some sense) most informative viewpoint. This is done by using only the first few principal components so that the dimensionality of the transformed data is reduced.

Lets see how is it applied to face recognition :
Every face image is represented as a huge matrix in digital world or computer, processing the images with that huge matrices is practically impossible, dimensions of the images has to be reduces, so PCA is applied to reduce the dimensions.

Every image a matrix of n*m is converted to a vector of 1*(n*m) dimensions. And this vector is taken through the algorithm of PCA and finally we extract, the same number of feature vectors as the number of original images and the extracted feature vector is much less in dimension and processing then is very easy in the real time.

1.1 ALGORITHM OF PCA

1.1.1 Get the high dimensional data

1.a) In our case images are the data, individual images are treated as a matrix of order m X n by the computer, these matrices are reshaped into a vector of size (m*n) X 1 (Note : here m is the number of rows and n is the number of columns)
1.b) Step 1.a is repeated to all the images in the database
1.c) Now all the vectors are put together to form a huge matrix of the entire database of the order m*n X 1, lets give a name as X

1.1.2 Calculate the mean image vector (m)

Note : we can calculate this by adding all the vectors and dividing by no of images in database

1.1.3 Calculate the adjust matrix (XA = X-m)

This calculated by subtracting each vector by the mean image vector

1.1.4 Calculate the co-variance matrix (Cov=XA*XA)

1.1.5 Calculate the eigen values and eigen vectors of the obtained co-variance matrix

1.1.6 Multiply the XA (adjust matrix) and the eigen vectors to get the mapping / signature of the original images/vectors, lets call this value as cov-eig

1.1.7 Select the leading eigen values in all the vectors of cov-eig

1.1.8 Finally Multiply the transpose of XA (adjust matrix) with the leading eigen values selected in the provious step, this value will be of the order (no of images in database) X (no of leading eigen values selected), which is the final matirix (signature of the entire database)