CSE 4835
Pattern Recognition Lab
Lab Report 05

Implementing a feature selection system using PCA.

Submitted by:
Md. Redwan Karim Sony
Student ID: 124401

Task 01:

task is to carry out Principal Component Analysis (PCA) on a set of face images. The basic task is to learn the use of *princomp()* MATLAB function in order to obtain the Eigen-vectors and Eigen-values. Reducing the original feature dimension is the main objective.

Procedure:

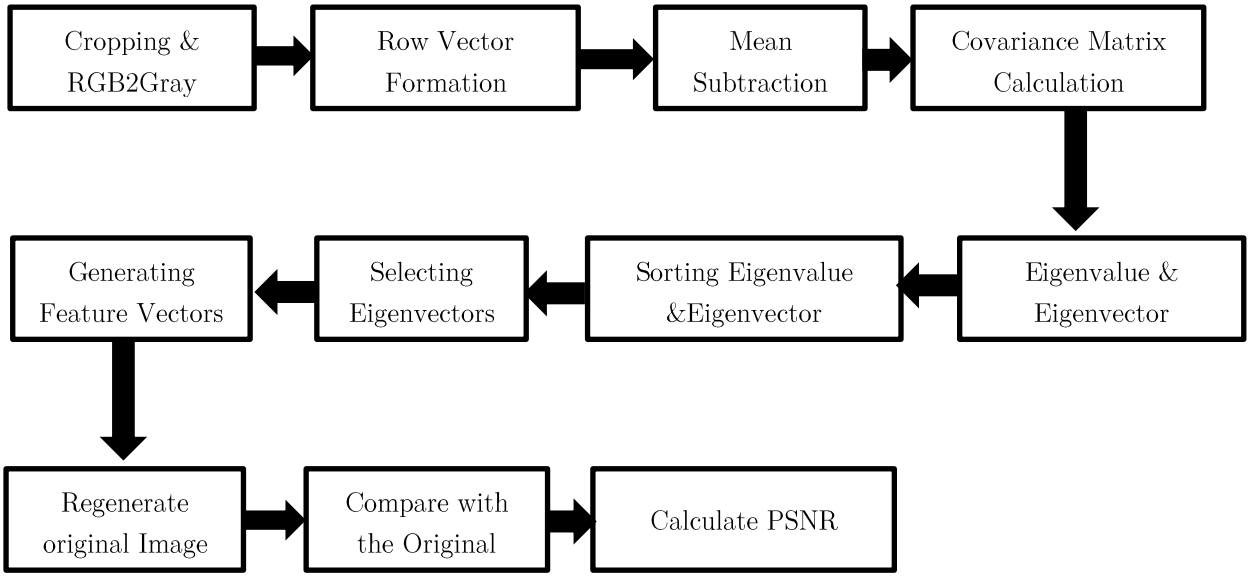


Figure 1: Flow Diagram of the feature selection, feature reduction and PSNR calculation of an image.

- **Cropping and RGB2Gray Conversion:** At first all the image are loaded and they are converted from RGB to Grayscale and they are cropped into predefined shape of 50x50 pixel size.
- **Row Vector Formation:** Now each of the pixel of the image is converted from a matrix to a one dimensional array and put to a matrix of which, each row represents an individual image.
- **Mean Subtraction:** Mean of the raw matrix formed is calculated and subtracted from each of the images.
- **Covariance Matrix Calculation:** The covariance matrix is calculated using ***cov()*** function of ***MATLAB.***

- **Eigenvalue and Eigenvector Calculation:** Then normally we use ***eig()*** function of ***MATLAB*** in order to get the eigenvalue and eigenvectors.

- **Eigenvectors Selection:** In this step we sort the eigenvalues and eigenvectors in a non-decreasing order. Then we select the suitable number of the eigenvectors which represent the feature space.
- **Generating the feature vector:** In this stage we create a matrix using each of the eigenvectors as column vector. This is our feature space. This follows the following formula

  FinalData = RowFeatureVector * RowDataAdjust ----------(1)

  Where,

  FinalData = features;

  Row FeatureVector = matrix formed by eigenvectors

  RowDataAdust = Mean subtracted raw data

- **Regenerate Original Image:** In order to regenerate the original image we have to do the following operation.

  $RawDataAdjust = (RowFeatureVector)^T * FinalData$ ---------(2)

  Where,

  FinalData = features;

  Row FeatureVector = matrix formed by eigenvectors

  RowDataAdust = Mean subtracted raw data

- **Comparison:** In this state each of the row of ***FinalData*** represents each of the image in the reduced feature space. Now using the equation (2) we regenerate each of the image in the reduced feature space. And from the ***RawDataAdjust*** we can generate the original image.

- **PSNR Calculation:** **(Peak signal-to-noise ratio, PSNR) is calculated by the following formula.**

$$PSNR = 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE)$$

Where,

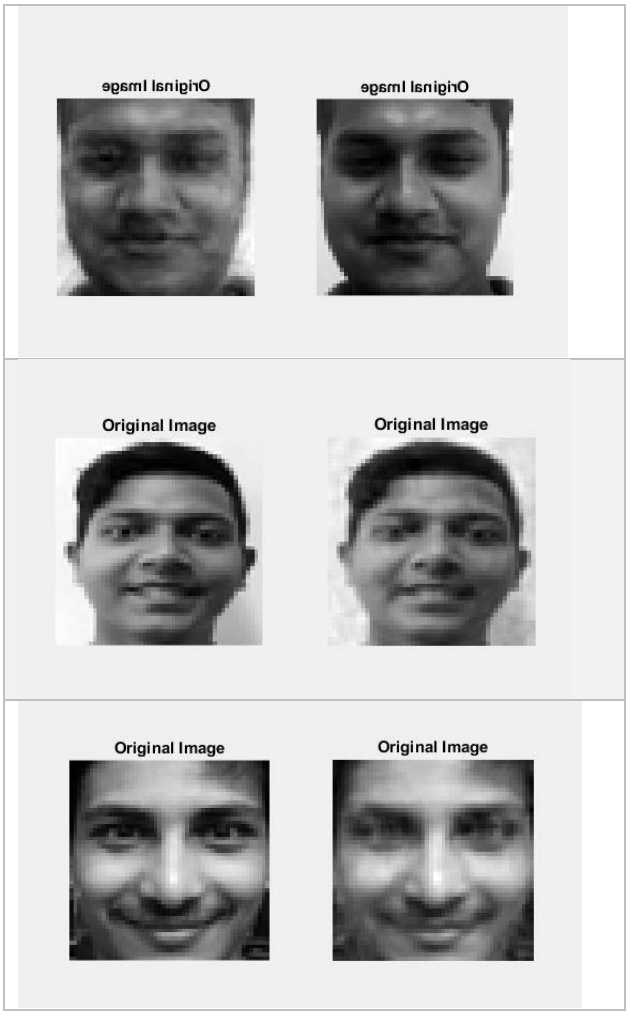
MAX_I = maximum intensity value of the original image and

$$MSE = \frac{1}{m\,n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$$

where I and K are the original and regenerated images.

Example:

Some of the image which have been processed are shown below where both original and reconstructed image are side by side



Findings and Discussion:

- Here this process can be performed using ***princmp()*** function of ***MATLAB***.
- The more we omit the eigenvectors with corresponding lower eigenvalues the more we lose information.
- If we remove some of the eigenvectors with non-zero eigenvalues, the we can never get 100% original image back.
- The optimal range of **PSNR** value is greater than 30. The more is the **PSNR** the better is the compression i.e. less is the noise corresponding to the signal strength.

- This PCA method can be used to compress a group of image while storing.
- PCA method uncorrelates the data and reduce the dimension.
- With the help of PCA, we can get rid of curse of dimensionality while working with the data that are of many features.
- Each of the eigenvectors are orthogonal and perpendicular to each other in their feature space. And the eigenvector corresponding to the maximum value of eigenvalue is called the principal component of the data distribution.

Associated MATLAB File:

- *PCA_BuiltInMethod.m (using **printcmp()**)*
- *PCA_Manual.m (using **eig()**)*