

# PCA Algorithm and its application in image compression and face recognition

Jitendra Suthar

M. Tech. Scholar

Department of Computer Science and Engineering

Jodhpur Institute of Engineering & Technology

Jodhpur, Rajasthan

[jitendra.suthar@jiertjodhpur.com](mailto: jitendra.suthar@jiertjodhpur.com)

Ashutosh Vyas

Associate Professor

Department of Computer Science and Engineering

Jodhpur Institute of Engineering & Technology

Jodhpur, Rajasthan

[ashutosh.vyas@jiertjodhpur.com](mailto: ashutosh.vyas@jiertjodhpur.com)

**Abstract**—PCA (Principal Component Analysis) is an algorithm which is mainly used for face detection and recognition. This algorithm is basically used for learning of sample (training) face images because before starting to face detection and recognition we have to require a set of Eigen values and Eigen vector of sample (trained) images. So PCA is used for face recognition. PCA is a method which is applied on the group of colour images.

This image is divided to some rows. The division of rows is according to variation in the image. Each row acts as an input for simple PCA algorithm. If the rows have the high data similarity then we apply HOTELLING transform on the rows of a given image. First we have to apply algorithms which find out the variation between the rows so we can divide the image into number of rows for PCA calculation. For the best result we find out the covariance value between the rows of image matrix.

**Index Terms**— Principal Component, Eigen value, Eigen vector, covariance matrix, HOTELLING transform, Standard Deviation, Variance, Covariance.

## I. INTRODUCTION

PCA (Principal Component Analysis) is an Algorithm which is mainly used for face detection and recognition. This algorithm is basically used for learning of sample (training) face images because before starting to face detection and recognition we have to require a set of Eigen values and Eigen vector of sample (trained) images. PCA (Principal Component Analysis) Algorithm basically takes sample images as an input and calculates the Face vector for each input image and then find out the average Image Face vector for all input Face vectors and then calculate the covariance matrix for all input (sample) images. For finding the Eigen value and Eigen vector of covariance matrix. A PCA (Principal Component Analysis) Algorithm is implemented by following steps-

1) First it takes sample images as an input.

- 2) Calculate Face vector for each input image.
- 3) After Finding the Face vectors it find out the Average Face vector of all input images Face vector.
- 4) Calculate the Covariance matrix of all input Face vectors.
- 5) Calculate Eigen value and Eigen vector of Covariance matrix.

Principal Component Analysis technique involve a mathematical procedure that transform a number of correlated variables (data) into a smaller number of uncorrelated variables (data). These transformed data is known as Principal Component of PCA Algorithm. First Principal component measures as much as variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible. So it is a technique of identifying patterns in data, and represents the data in such a way as to highlight their similarities and differences. In a high dimension data it is hard to find the pattern and luxury of graphical representation is not available, PCA is a powerful tool for analysing data.

The other main advantage of PCA is that once you have found these patterns in the given data, then you can compress the data by reducing the number of dimension, without much loss of information.

## II. PCA COMPUTATION

For implementing the PCA Algorithm using Covariance matrix follow these steps-

- 1. Get some data:** - Get the data set on which you want to apply PCA computation. This step takes the input data for which you want to implement PCA algorithm which is complex in nature.

**2. Subtract the mean:** - After getting the data, you have to subtract the mean from each of the dimensions. The mean value is subtracted from each dimension it means first calculate mean value along with x - dimension and y- dimension. Then find out the subtract of mean from every point this subtract value in PCA is known as variability

**3. Calculate the deviation from the mean value:** - To calculate the standard deviation square the difference calculated above then add all these after that divide this by n-1 and take square root of final result.

**4. Calculate covariance matrix:**-Now Calculate the covariance between each dimensions then put all these values in matrix.

**5. Calculate Eigen values and Eigenvector:** - If the covariance matrix is a square then it is easy to calculate the eigenvector and eigen value for given covariance matrix. Basically eigenvector of a matrix represent the lines that characterize the data. For any further calculation this calculated (eigenvector) data is used.

**6. Choosing components and forming a feature vector:** - This phase represents the data compression and reduced dimensionality. Eigenvalues are quite different values from Eigenvector. Eigenvector with the highest Eigenvalue is the principle component of the data set. It is the most important relationship between the data dimensions.

First eigenvectors are found from the covariance matrix then in next step is to order them by eigenvalue, highest to lowest. This gives components in order of significance of data. Now we decide to ignore the components of lesser significance. This loses some information, but if the eigenvalues are small, we do not lose much. If we leave out some components, the final data set will have lesser dimensions than the original.

To be precise, if our originally data have n dimensions and then we calculate n eigenvectors and eigenvalues then we choose only the first p eigenvectors so the final data set has only p dimensions. Now we calculate a feature vector, which is constructed by taking the eigenvectors that we want to keep from the list of Eigenvectors Feature Vector = (eig1, eig2, eig3.....eig N)

**7. Deriving the new data set:**-This is the last step of PCA. Once we have calculated and choose the components (eigenvectors) that we want to keep in our data and then create a feature vector. After that we take the transpose of the given vector and multiply it on the left of the original data set.

$$\text{Final Data} = \text{Row Feature Vector} * \text{Row Data Adjust}$$

where "row feature vector" represent the matrix with the eigenvectors in the columns transposed so that the eigenvectors are now comes in the rows, with the most principal Component eigenvector at the top, and "Row Data Adjust" is the mean-adjusted data transposed. For Example data items are in each column, with each row holding a separate dimension. "Final Data" set is the final data set, with the data items in columns and dimensions along rows.

### III. BACKGROUND MATHEMATIC FOR PCA ALGORITHM

**1. Statistics:** - Statistics is based on the idea that we have big set of data, and we want to analyze that set in terms of the relationships between the individual points in given data set. So this type of ideas comes in field of Statistics.

**2. Standard Deviation:** - The mean of sample which is simply defined by Add up all the numbers and then divide by how many numbers are. The mean value does not tell us a lot about the data except for a sort of middle point. The Standard Deviation (SD) of a data set is the measure that data is how much spread. It can be defines as the average distance from the mean of the data set to a point.

**3. Variance:** - Variance is same as Standard Deviation it is basically another measure of the spread of data in a dataset. In fact it is almost identical to the standard deviation. This is simply the square value of standard deviation (SD).

**4. Covariance:** - Standard deviation and variance two measures are purely 1-dimensional .Many data sets have more than one dimension and the aim of the statistical analysis of these datasets is usually to see if there is any relationship between the dimensions. Standard deviation and variance only operate on one dimension, so these only calculate the standard deviation for each dimension of the data set independently of the other dimensions.

However, it is useful to have a similar measure to find out how much the dimensions vary from the mean with respect to each other .Covariance is such a measure. Covariance is always measured between two dimensions. If you calculate the covariance between one dimension and itself, you get the variance.

**5. Covariance Matrix:** - A useful way to get all the possible covariance values between all the different dimensions is to calculate them all and put them in a matrix. For Example the entry on row 2, column 3, is the covariance value calculated between the 2<sup>nd</sup> dimension and the 3<sup>rd</sup> dimension.

**6. Matrix Algebra:** - This section serves to provide a background for the matrix algebra required in PCA. Specifically I will be looking at eigenvectors and eigenvalues of a given matrix.

**7. Eigenvectors:** - In mathematics, eigenvalue, eigenvector, and Eigenspace are related concepts in the field of linear algebra. Eigenvalues, eigenvectors and Eigenspace are properties of a matrix. In general, a matrix acts on a vector by changing both its magnitude and its direction. However, a matrix may act on certain vectors by changing only their magnitude, and leaving their direction unchanged these vectors are the eigenvectors of the matrix. Formally, if  $A$  is a linear transformation, a non-null vector  $X$  is an eigenvector of  $A$  if there is a scalar  $\lambda$  such that  $AX = \lambda X$ .

The scalar  $\lambda$  is said to be an eigenvalue of  $A$  corresponding to the eigenvector  $X$ . Lastly, all the eigenvectors of a matrix are perpendicular, i.e. at right angles to each other, no matter how many dimensions you have. By the way, another word for perpendicular, in maths talk, is orthogonal. This is important because it means that you can express the data in terms of these perpendicular eigenvectors, instead of expressing them in terms of the x and y axis. We will be doing this later in the section on PCA.

**8. Eigen value:** - A matrix acts on an eigenvector by multiplying its magnitude by a factor, which is positive if its direction is unchanged and negative if its direction is reversed. This factor is the eigenvalue associated with that eigenvector. An Eigenspace is the set of all eigenvectors that have the same eigenvalue, together with the zero vectors.

**9. Eigen faces:** - Eigen faces are a set of eigenvectors used in the computer vision for human face recognition. These eigenvectors are derived from the covariance matrix of the probability distribution of the high-dimensional vector space of possible faces of human beings. A set of Eigenfaces can be generated by applying a mathematical process called principal component analysis on a large set of input images depicting different human faces. Eigenfaces can be considered a set of standardized face ingredients, derived from statistical analysis of many input images of faces.

Any new captured human face can be considered to be a combination of these standard input faces. In PCA method a person's face is not recorded by a digital photograph, but instead as just a list of values which are much less space is taken for each person's face. The Eigenfaces that are created will appear as light and dark areas that are arranged in a particular pattern. This pattern calculates that how different

features of a face are singled out to be evaluated and scored. Also, there will be a pattern to evaluate symmetry for input faces, if there is any style of facial hair, where the hairline is, or evaluate the size of the nose or mouth. Other Eigenfaces have patterns that are less simple to identify, and the image of the Eigenfaces may look very little like a face. This technique is also used for handwriting analysis, lip reading, voice recognition, sign language/hand gestures interpretation and medical imaging analysis.

#### IV. METHOD OF IMAGE COMPRESSION BY USING PCA ALGORITHM

PCA is method is applied on the group of images now we apply PCA to an image and the image is a colored image. this image is divided to some rows. The division of rows is according to variation in the image. Each row acts as an input for simple PCA algorithm. If the rows have the high data similarity then we apply HOTELLING transform on the rows of an given image. First we have to apply an algorithm which finds out the variation between the rows so we can divide the image into number of rows for PCA calculation. For the best result we find out the covariance value between the rows of image matrix.

Covariance between two random variable is

$$\text{cov}(a,b) = E[(a - \mu_a)(b - \mu_b)]$$

where  $\mu_a = E[a]$ ,  $\mu_b = E[b]$

For this we find out similarity between the rows of an image matrix

$$\text{similarity}(i) = \text{cov}(i \text{ row}; i + 1 \text{ row})$$

where

$$i = 1, 2, 3, 4, \dots, n-1$$

take the two neighbor rows of an given image matrix as  $a$  and  $b$  rows and find out the similarity between these rows if similarity is very small then leave this row and find out similarity between first and third row and so on calculate the similarity between rows and discard the rows.

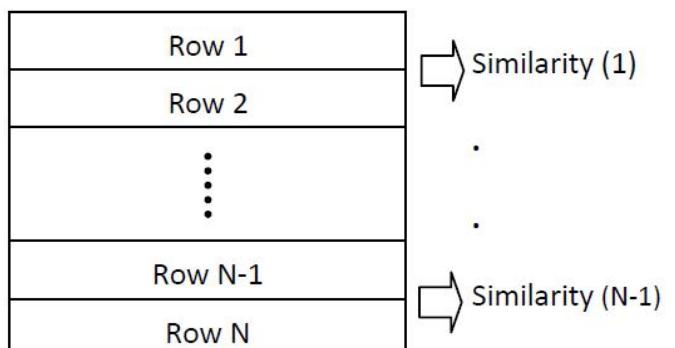


Figure 1: The rows of  $M * N$  matrix and their similarity values

In order to improve the efficiency of mentioned algorithm to divide image into some rows, we generate a curve of similarity values between its rows of an given image. Suppose there are  $M$  square  $N \times P$  monochrome images. Colored images can be supposed as 3 monochrome images that are red, green, and blue color components for each individual pixel  $N \times P$  monochrome images are equivalent to  $N \times P$  matrixes that the values of the components of the matrixes corresponding pixel's location. Suppose  $N \times P = Q$ . By reshaping the matrixes, the image can be expressed as  $1 \times Q$  vectors  $F_i$  in equation 1. All images are put in  $X$  matrix that its elements are the intensity values of input images.

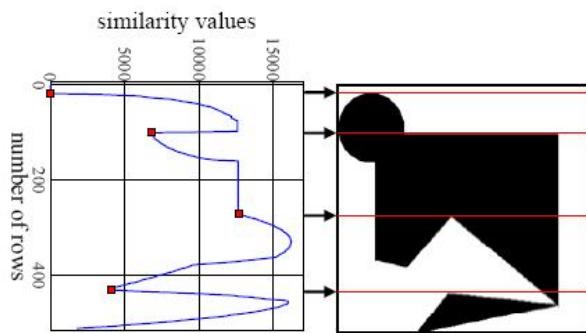


Figure 2: A simple test image and the similarity value curve of the image.

#### REFERENCES

- [1] Zhaodi Xiao, School of Sciences, South China University of Technology, Guangzhou, China "An Image Compression Algorithm Based on PCA and JL" Journal of Image and Graphics Vol. 1, No. 3, September 2013
- [2] Farshid Mehrdoust "A hybrid randomized algorithm for image compression" International Journal of Applied Mathematical Research, 2 (1) (2013) 1-7
- [3] S. Zebhi, M. R. Aghabozorgi Sahaf, and M. T. Sadeghi "IMAGE FUSION USING PCA IN CS DOMAIN" Signal Processing Research Group, Electrical and Computer Engineering Department, Yazd University Signal Image Processing : An International Journal (SIPIJ) Vol.3, No.4, August 2012

- [4] R. Navaneethakrishnan "Study of Image Compression Techniques" International Journal of Scientific Engineering Research, Volume 3, Issue 7, ISSN 2229-5518, July-2012
- [5] Mostafa Mofarreh-Bonab "A New Technique for Image Compression Using PCA" University of Bonab, East Azerbaijan, Iran 2012
- [6] Sherin Kishk, Hosam Eldin Mahmoud Ahmed and Hala Helmy "Integral Images Compression using Discrete Wavelets and PCA" International Journal of Signal Processing, Image Processing and Pattern Recognition Vol. 4, No. 2, June, 2011
- [7] Pramod Kumar Pandey, Yaduvir Singh, Sweta Tripathi "Image Processing using Principle Component Analysis" International Journal of Computer Applications Volume 15 □ No:4 , February 2011
- [8] Stefan Stolevski "Hybrid PCA Algorithm for Image Compression" 18<sup>th</sup> Telecommunications forum TELFOR 2010.
- [9] Prof. Y. Vijaya Lata, Chandra Kiran Bharadwaj Tungathurthi, H. Ram Mohan Rao, Dr. A. Govardhan, Dr. L. P. Reddy "Facial Recognition using Eigenfaces by PCA". International Journal of Recent Trends in Engineering, Vol. 1, No. 1, May 2009
- [10] Daoqiang Zhang, Zhi-Hua Zhou and Songcan Chen "Diagonal Principal Component Analysis for Face Recognition" National Laboratory for Novel Software Technology Nanjing University, Nanjing 210093, China 2006
- [11] M. Mudrova, A. Prochazka "PRINCIPAL COMPONENT ANALYSIS IN IMAGE PROCESSING" Institute of Chemical Technology, Prague Department of Computing and Control Engineering 2005
- [12] M. Turk and A. Pentland, "Eigenfaces for Recognition", Journal of Cognitive Neuroscience, vol. 3, no. 1, pp. 71-86, 1991
- [13] "Iterative Methods for Computing Eigenvalues and Eigenvectors" Maysum Panju University of Waterloo.
- [14] Raghuraman Gopalan, Prof. Shihab Shamma "IMAGE COMPRESSION AND FACE RECOGNITION USING PRINCIPAL COMPONENT ANALYSIS" ENEE 630: Advanced Digital Signal Processing
- [15] "Eigenfaces for Recognition" (M. Turk, A. Pentland)