

# ML | Face Recognition Using Eigenfaces (PCA Algorithm)

Difficulty Level : Hard • Last Updated : 24 Sep, 2021



In 1991, *Turk and Pentland* suggested an approach to face recognition that uses dimensionality reduction and linear algebra concepts to recognize faces. This approach is computationally less expensive and easy to implement and thus used in various applications at that time such as handwritten recognition, lip-reading, medical image analysis, etc. PCA (Principal Component Analysis) is a dimensionality reduction technique that was proposed by Pearson in 1901. It uses Eigenvalues and EigenVectors to reduce dimensionality and project a training sample/data on small feature space. Let's look at the algorithm in more detail (in a face recognition perspective).

## Training Algorithm:

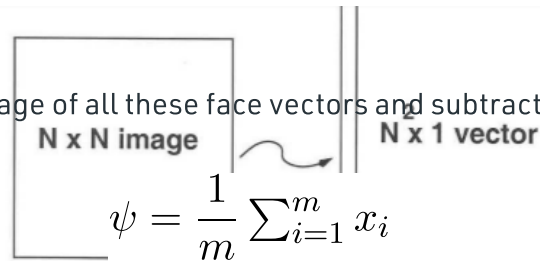
- Let's Consider a set of  $m$  images of dimension  $N \times N$  (training images).



Training Image with True Label (LFW people's dataset)

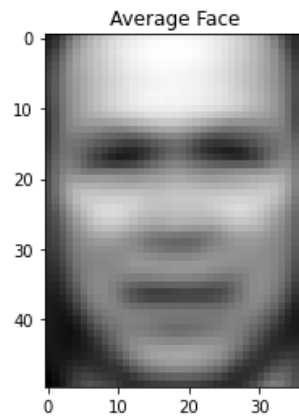
- We first convert these images into vectors of size  $N^2$  such that:

- Now we calculate the average of all these face vectors and subtract it from each vector



$$\psi = \frac{1}{m} \sum_{i=1}^m x_i$$

$$a_i = x_i - \psi$$



average\_face

- Now we take all face vectors so that we get a matrix of size of  $N^2 * M$ .



$$A = [a_1 \ a_2 \ a_3 \ \dots \ a_m]$$

- Now, we find Covariance matrix by multiplying  $A$  with  $A^T$ .  $A$  has dimensions  $N^2 * M$ , thus  $A^T$  has dimensions  $M * N^2$ . When we multiplied this gives us matrix of  $N^2 * N^2$ , which gives us  $N^2$  eigenvectors of  $N^2$  size which is not computationally efficient to calculate. So we calculate our covariance matrix by multiplying  $A^T$  and  $A$ . This gives us  $M * M$  matrix which has  $M$  (assuming  $M \ll N^2$ ) eigenvectors of size  $M$ .

$$Cov = A^T A$$

- In this step we calculate eigen values and eigenvectors of above covariance matrix using the formula below.



$$C'u_i = \lambda_i u_i$$

where,

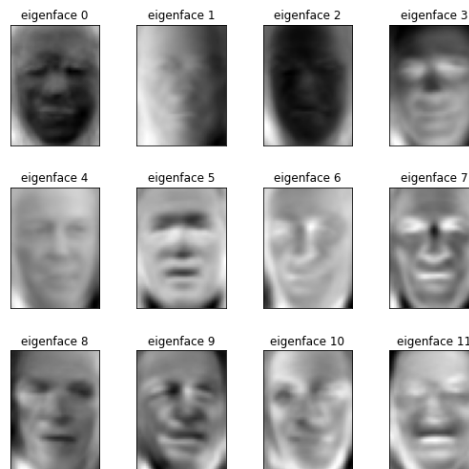
$$C' = AA^T \text{ and } u_i = A\nu_i$$

From the above statement It can be concluded that  $C'$  and  $C$  have same eigenvalues and their eigenvectors are related by the equation  $u_i = A\nu_i$ . Thus, the  $M$  eigenvalues (and eigenvectors) of covariance matrix gives the **M** largest eigenvalues (and eigenvectors) of  $C'$

- Now we calculate Eigenvector and Eigenvalues of this reduced covariance matrix and map them into the  $C'$  by using the formula  $u_i = A\nu_i$ .
- Now we select the  $K$  eigenvectors of  $C'$  corresponding to the  $K$  largest eigenvalues (where  $K < M$ ). These eigenvectors has size  $N^2$ .
- In this step we used the eigenvectors that we got in previous step. We take the normalized training faces (face – average face)  $x_i$  and represent each face vectors in the linear of combination of the best **K** eigenvectors (as shown in the diagram below).

$$x_i - \psi = \sum_{j=1}^K w_j u_j$$

These  $u_j$  are called **EigenFaces**.



EigenFaces

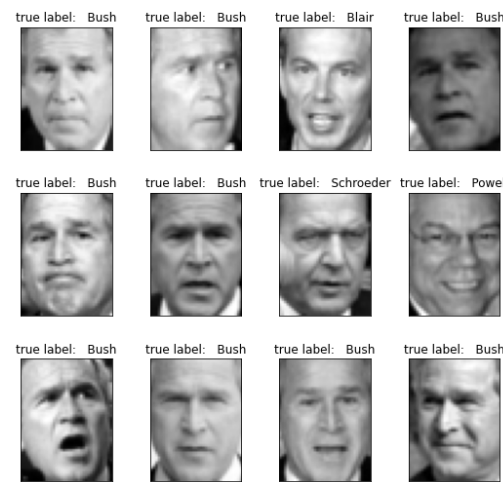
- In this step, we take the coefficient of eigenfaces and represent the training faces in the form of a vector of those coefficients.

$$x_i = \begin{bmatrix} w_1^i \\ w_2^i \\ w_3^i \\ \vdots \\ w_k^i \end{bmatrix}$$



Linear Combination of EigenFaces

### Testing/Detection Algorithm :



Test Images With true labels

- Given an unknown face  $y$ , we need to first preprocess the face to make it centered in the image and have the same dimensions as the training face
- Now, we subtract the face from the average face  $\psi$ .

$$\phi = y - \psi$$



Test Images - Average Images

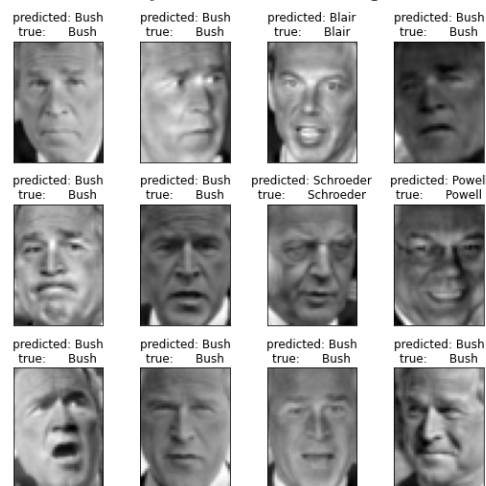
- Now, we project the normalized vector into eigenspace to obtain the linear combination of eigenfaces.

$$\Omega = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_k \end{bmatrix}$$

- We take the vector generated in the above step and subtract it from the training image to get the minimum distance between the training vectors and testing vectors

$$e_r = \min_l \|\Omega - \Omega_l\|$$

- If this  $e_r$  is below tolerance level  $T_r$  then it is recognised with  $l$  face from training image else the face is not matched from any faces in training set.



Test images With prediction

## Advantages:

- Easy to implement and computationally less expensive.
- No knowledge (such as facial feature) of the image required (except id).

## Limitations :

- Proper centered face is required for training/testing.
- The algorithm is sensitive to lightning, shadows and also scale of face in the image .
- Front view of the face is required for this algorithm to work properly.

## Reference :

- [Turk and Pentland's 1991 paper on face recognition](#)