PCA on Face Images and Face Reconstruction

The main idea of principal component analysis (PCA) is to reduce the dimensionality of a data set consisting of many variables correlated with each other, either heavily or lightly, while retaining the variation present in the dataset, up to the maximum extent.

PCA Algorithm

1. Multi-dimensional Data is stored in a matrix **X**.
2. Two *variables* are *correlated* if there is a linear relationship in between them. To study these dependencies between variables, we use a covariance matrix.
3. for any covariance matrix **Σ** there exists matrix **V** called eigenvector matrix and *diagonal* matrix **Λ** called eigenvalue matrix, such that the expressions below hold.

$$\Sigma_X = (X - \mu_X)^T (X - \mu_X)$$
$$\Sigma_X V = V\Lambda$$
$$V^T V = E$$

4. PCA normalizes the data matrix **X** to zero mean and then multiples by *some matrix* **P**. The multiplication is actually linear transformation of data. That means if we choose **P** very carefully, we can either *rotate*, *scale* or *project* the data into vector subspace.

$$Z = PCA(X) = (X - \mu_X)P$$

$$\Sigma_Z = [(X - \mu_X)P]^T[(X - \mu_X)P]$$
$$\Sigma_Z = P^T(X - \mu_X)^T(X - \mu_X)P$$
$$\Sigma_Z = P^T\Sigma_X P$$

5. There is a relationship between both covariance matrices of **X** and **Z**, and if we choose **P** to be eigenvector matrix **V** defined above we get -
6. This means that the projected matrix **Z** is uncorrelated and its variables have no longer any kind of linear dependency [because **Λ** is diagonal matrix].

PCA on images and Face Reconstruction

Steps taken:

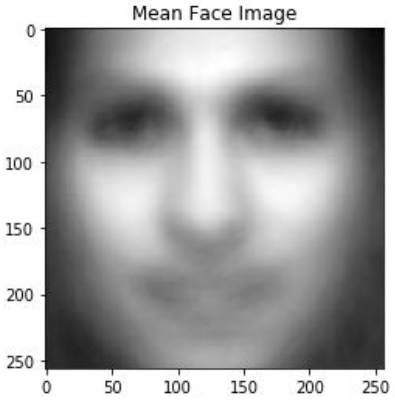
- **Reading images** which is N x N (here 256 x 256) as a vector of dimension (N*N) x 1. Dataset contains **m** images (here m = 520). Then matrix **X** = **m** x (**N* N**).

- Now **X** contains training faces I_1, I_2, I_3, ……I_m images. Example is shown in figure below.

Training face images

- Every image I_i is represented as a vector Γ_i
- Calculated the mean image of the given dataset Ψ :

$$\Psi = \frac{1}{m} \sum_{i=1}^{m} \Gamma_i$$



- Subtract the mean face from each faces :

$$\varphi_i = \Gamma_i - \Psi$$

- Compute the covariance matrix C :

$$C = \frac{1}{m} \sum_{n=1}^{m} \varphi_n \varphi_n^T = XX^T \quad (N^2 \times N^2 \text{ matrix})$$

Here **N** is 256 so (65536 x 65536 matrix) which is computationally not feasible.

$A^T A v_i = \mu_i v_i \Rightarrow AA^T A v_i = \mu_i A v_i \Rightarrow$

$CAv_i = \mu_i Av_i$ or $Cu_i = \mu_i u_i$ where $u_i = Av_i$

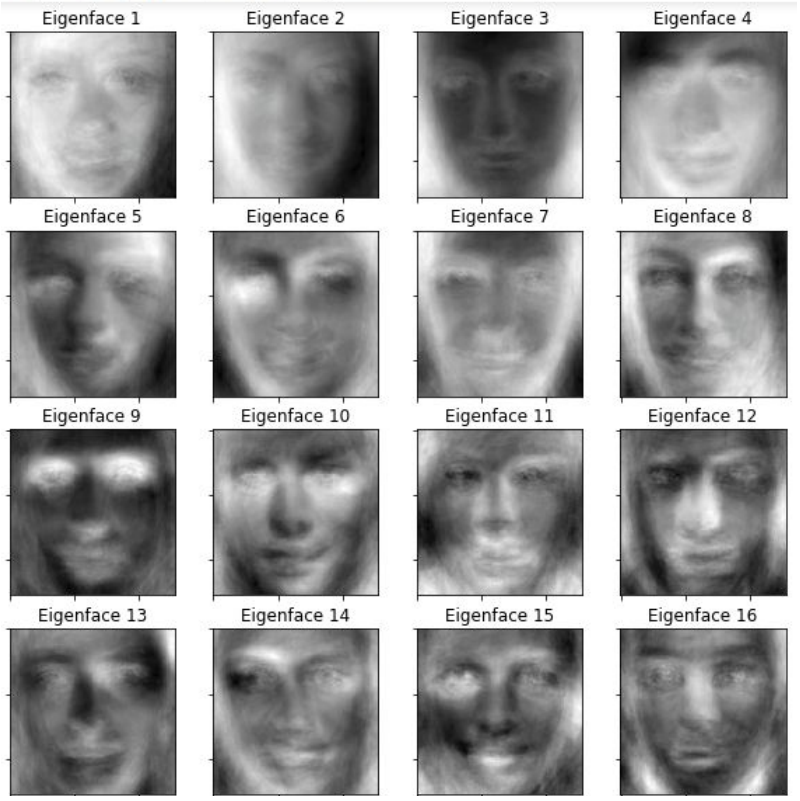
Thus, AA^T and $A^T A$ have the same eigenvalues and their eigenvectors are related as follows: $u_i = Av_i$!!

- Consider the matrix $X^T X$ (m x m matrix). We know from the above derivation that $X^T X$ and XX^T will have same eigenvalues and eigenvectors are related as given in equation above. Hence, we can calculate covariance matrix in feasible manner by this method and the corresponding eigenvectors and then convert them to get the eigenvectors of XX^T.
  XX^T can have up to N^2 eigenvectors and $X^T X$ can have m eigenvectors, so these m eigenvectors corresponds to the largest m eigenvectors of XX^T.

- Compute m eigenvectors of the XX^T by using the m eigenvectors of $X^T X$:
  $\Lambda_i = X\nu_i$

- Normalize eigenvectors Λ_i such that its L-2 norm is 1.

- Keep only k largest eigenvectors (corresponding to the K largest eigenvalues). I am using k = 16 eigenvectors for further calculations. We call these eigenvectors as eigenfaces. Eigenfaces plotted for this given dataset is show below :



- Now we can represent any face using linear combination of these eigenfaces.

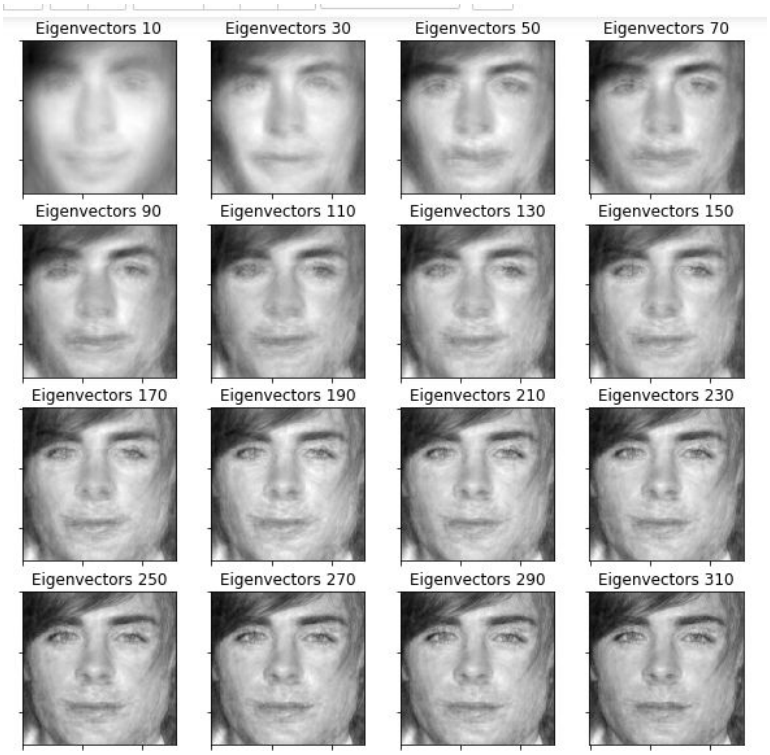
$$\widehat{\varphi}_i - mean = \sum_{j=1}^{k} w_j \Lambda_j$$

Where our w is weight matrix which is calculated as $w_j = \Lambda_j^T \varphi_i$

- Now we can represent each face φ with weight matrix i.e.

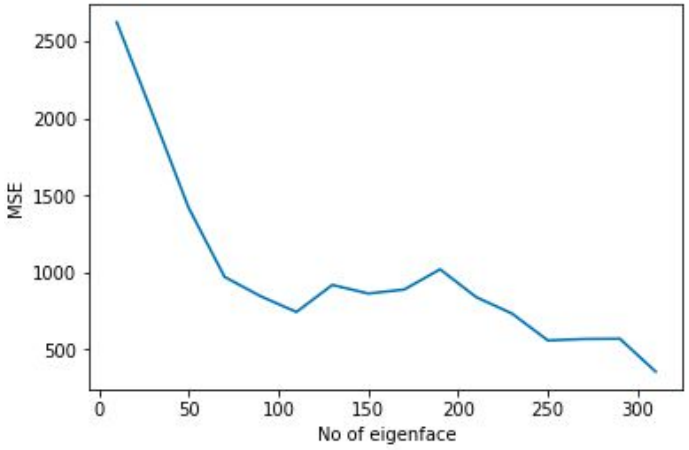
$$\varphi_i = \left[w_1\ w_2\ w_3\ w_4\ .....\ w_k\right]^T$$

- Example of face reconstruction on one of the image in dataset is show below.

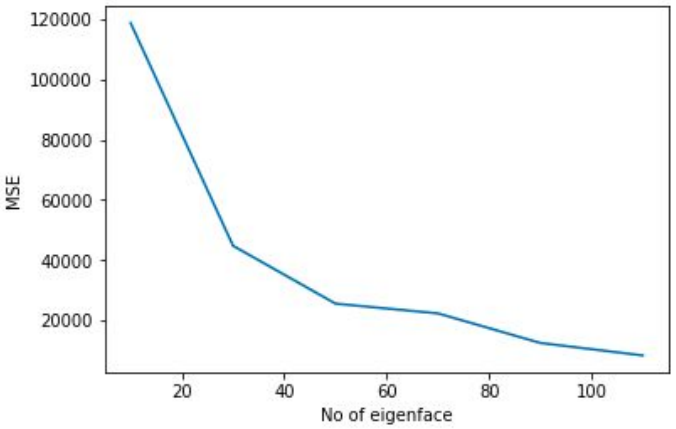


- As we can see on increasing count of eigenvectors/eigenfaces image becomes more and more clear. On taking 310 eigenvectors image is very close to the original image and hence, it Shows that we can reconstruct images using these eigenfaces.

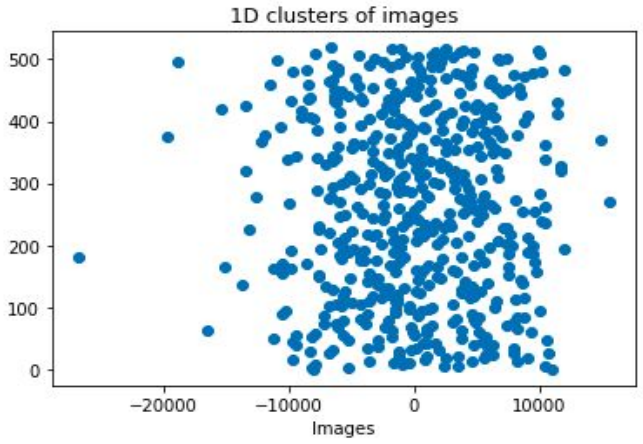
- For this similar image whose face reconstruction is shown, Mean Squared Error is plotted with count of eigenfaces.

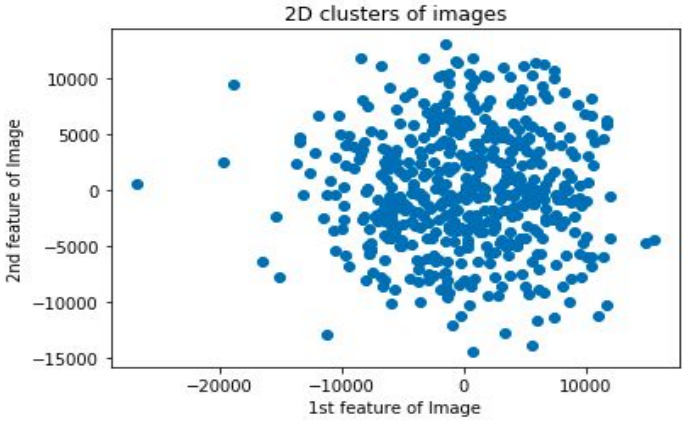


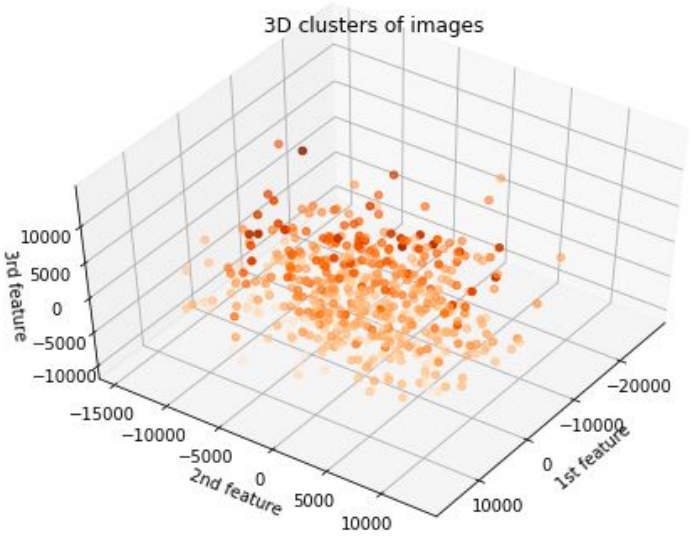
- Sum of Mean square error of some images with eigenfaces is plotted and shown below:



- Images are converted into 1D, 2D, and 3D using eigenfaces shown above and then these images are plotted to understand the distribution of images.







- These plots show that most of the images are near to the mean and as we move away from mean the count decreases. This type of distribution is similar to gaussian distribution.
  Hence, we can conclude that these images are in Gaussian Distribution, and we can use this assumption in our further experiments.