Report

All the images given in the dataset are of size 256 * 256 in dimension . So we can consider that each image has a total of 65536 dimensions . We can also consider that the each image has 65536 features . We can do PCA on this particular dataset in two ways :

First way :

Since while finding the covariance matrix the dimension will of the matrix will be 65536 * 65536 it will exceed memory constraints so we will first resize the image size from 256 * 256 to 32 * 32 . I have chosen this dimensions because the algo will work quicker if the dimension are less . Then according to the general pca find covariance matrix and get the top k eigenvalues and their corresponding eigenvectors to find the matrix U having dimensions k * 1024 (32 * 32) which will be used to compute the reduced dimensional vectors .

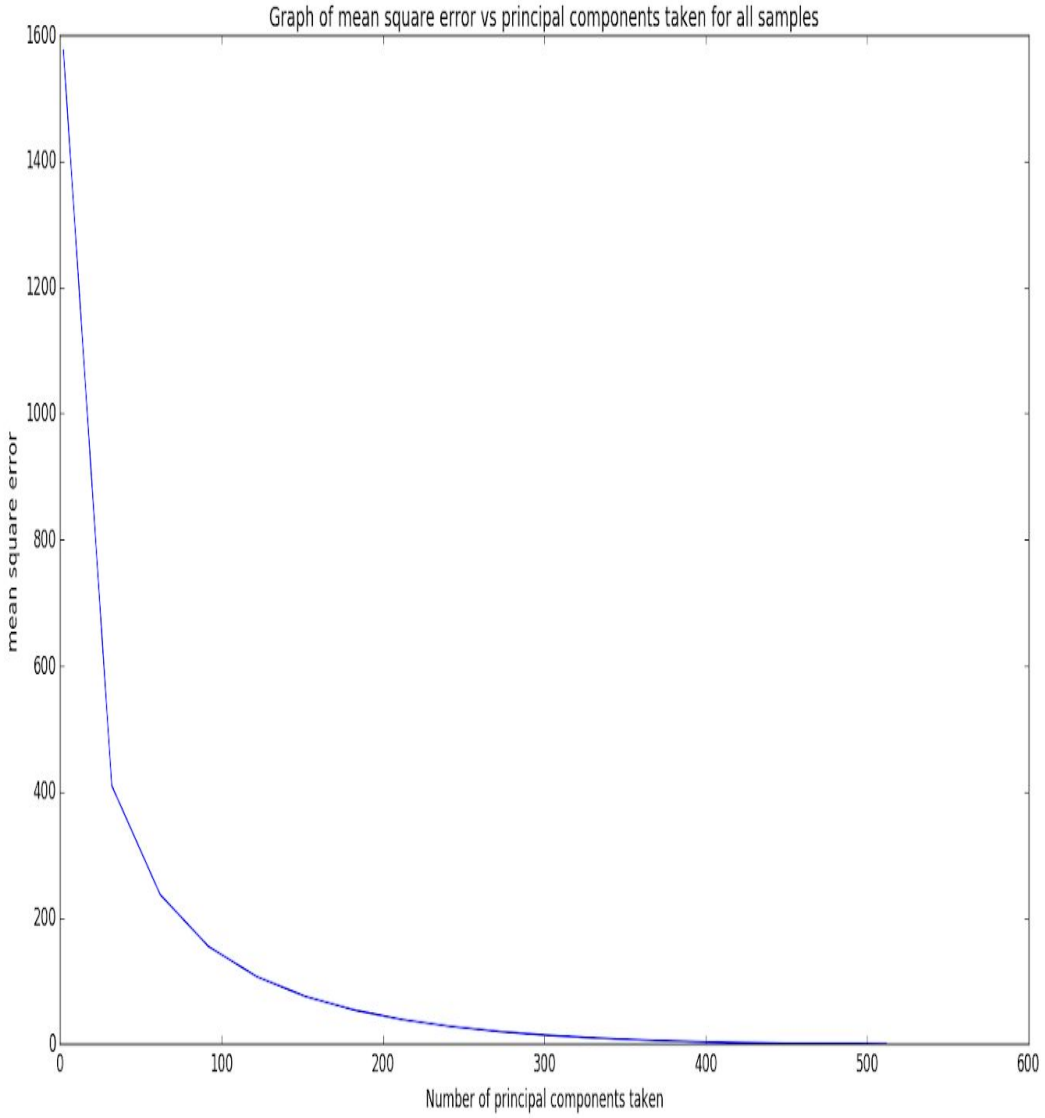
Second way :

Since in our case there are only 520 images i.e N = 520 and the d = 256 *256 instead of finding the eigenvectors for the covariance matrix xTx we can find the eigenvectors for the matrix xxT and from them we can get the eigenvectors corresponding to matrix xTx . This is explained below ,

xTx u = lambda * u
xxT v = lambda * v

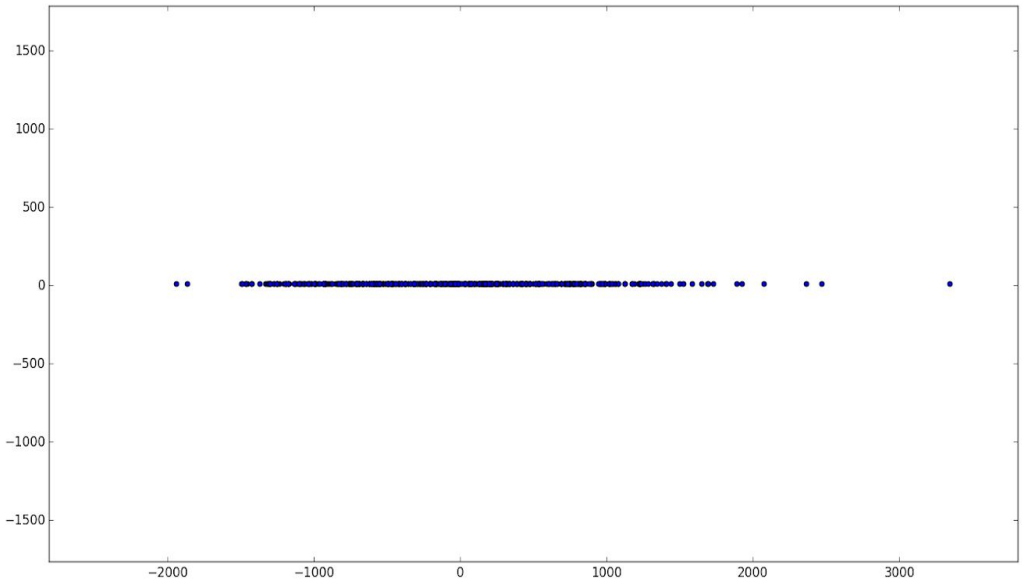
u = xTv
We will first find v's then from them we will find u's , then the process is same like above .

I have done the PCA in both the ways and the results were quite similar for finding the mean square error , here is the graph obtained for the number of principal components considered vs the mean square error for all the samples .

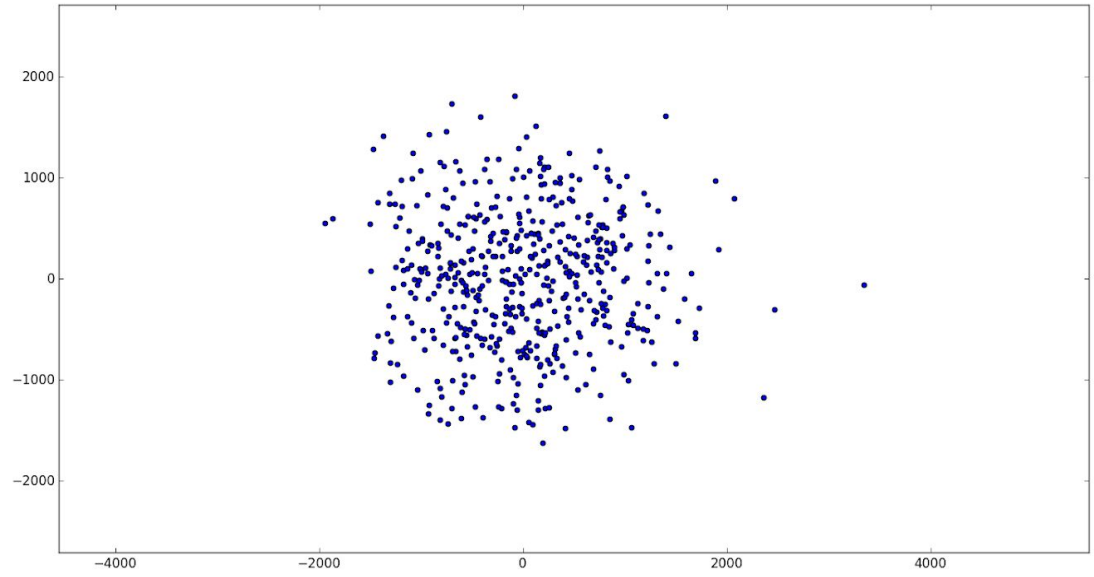


As we can see clearly from above as the number of principal components taken increases the mean square error decreases and almost approaches zero . This is because as more components are taken more will be the data we capture (more variance) .

Clustering of all the images on 1d scatter plot taking only one principal component



Clustering of all the images on 2d scatter plot taking two principal components



Clustering of all the images on 3d scatter plot taking three principal components