In this paper, the problem of facial recognition has been solved by considering faces as 2D images rather than complicating the data by considering the 3D intrinsic patterns normally used for recognition.
A face image has been considered as a point in a N^2 X N^2 dimensional space. Faces, being similar in overall configuration will not be randomly distributed in this high dimensional space and thus can be described by a relatively low dimensional sub-space.
We use PCA to find vectors that account for the distribution of face images over the entire image space. This sub-space is referred to as 'face-space'.

The procedure

- ***The DataSet -*** I have used an emoji's data set to perform the recognition. The training set involves images from 10 classes of faces with 4 images of each class. The test set has 10 images in total with each image belonging to a class. The images in each class vary according to the facial hair and accessories they wear.

- ***Finding the EigenFaces***

1. We first convert the faces to a 1D array and stack all of them vertically in a Matrix. This matrix is of shape (27000,40)
2. We find the average face by taking mean of all faces. This average face is then subtracted from all face images to make a mean subtracted matrix of face images. This is done so that all images are centred around 0 and with a standard deviation of 1
3. Our goal is to find the eigenvectors of the covariance matrix found by multiplying the mean subtracted matrix with its transpose.

$$C = \frac{1}{M} \sum_{n=1}^{M} \Phi_n \Phi_n^T \qquad A = [\Phi_1 \ \Phi_2 \ . \ . \ . \ \Phi_M].$$

$$= AA^T$$

4. The covariance matrix will be a N^2 X N^2 matrix. Finding the eigenvectors of a 27000 X 27000 matrix is a computationally expensive job and also if the number of data points or images is less than N^2, there will only exist M-1 meaningful eigenvectors. (M = number of images in train set)
5. We indirectly find the eigenvector by computing the eigenvectors of

$$L = A^T A, \text{ where } L_{mn} = \mathbf{\Phi}_m^T \mathbf{\Phi}_n,$$

6. Then we take the linear combinations of these eigenvectors with the images to produce the eigenvectors of the main covariance matrix. Below, u1, u2, u3 ,..u40 are the 40 eigenvectors of the 27000X27000 covariance matrix. These vectors are called "*eigenfaces*". We only consider 10 eigenvectors with the highest values as our principal components.

for example,

$M = 40$ images

$A^T A = 40 \times 40$ Covariance matrix ~~matrix~~ of mean subtracted images

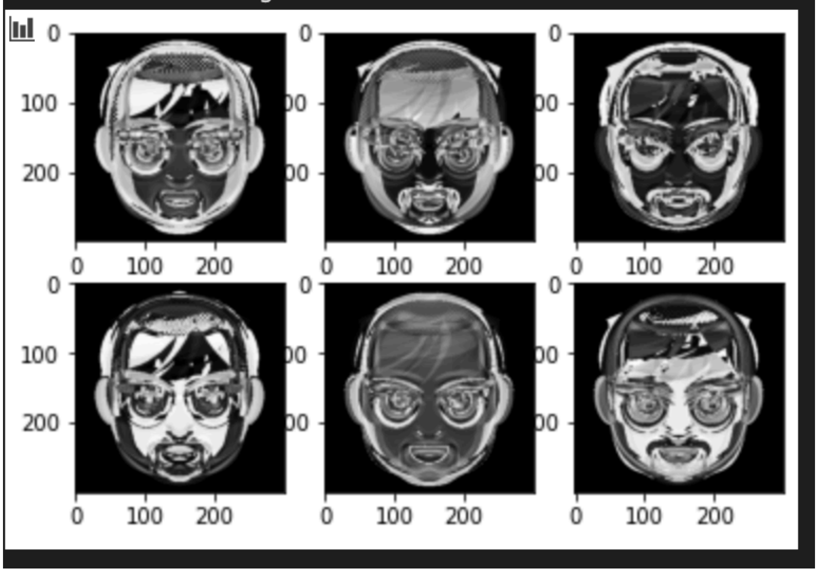
$$u_l = \sum_{k=1}^{40} v_{lk} \phi_k \qquad l = 1, \cdots 40$$

$$\Rightarrow u_1 = \sum_{k=1}^{40} v_{1k} \phi_k$$

$$u_2 = \sum_{k=1}^{40} v_{2k} \phi_k$$

$\vdots$

Below are some eigenfaces



- ***Applying recognition to a new face -***

1. A new face is converted into to its eigenface representation by taking the dot product of the mean subtracted image with the eigenvector. This is done with all the 10 eigenvectors we have chosen. We get a new vector with these 10 elements, lets call it the "*weight vector*"

$$\omega_k = \mathbf{u}_k^T(\mathbf{\Gamma} - \mathbf{\Psi})$$

$$\mathbf{\Omega}^T = [\omega_1, \omega_2 \ . \ . \ . \ \omega_{M'}]$$

2. We find the weight vectors for all the classes of images similarly and take the average weight vector as we have 4 images of each class.

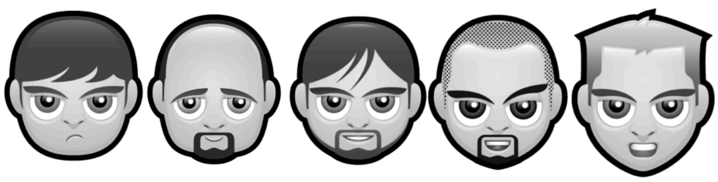
3. Using the euclidean distance we find the face class which is closest to the new image.

$$\epsilon_k = \|(\mathbf{\Omega} - \mathbf{\Omega}_k)\|^2$$

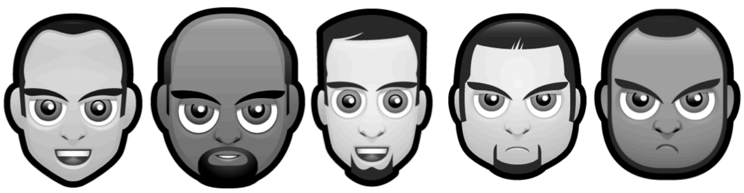
Results

With the emoji's data set, I have been able to produce an **accuracy of 80%** in face recognition

Classes 1-5



Classes 6-10



Wrongly Predicted images

Image	prediction	Original class image