# Face Recognition Using Eigenfaces

Athul P
*B170509EC*
*National Institute of Technology Calicut*
athul_b170509ec@nitc.ac.in

## I. Aim

1) Implement the Face recognition using eigenfaces paper by MA Turk.
2) Train the system using Yale face dataset
3) Calculate the accuracy of the system.

## II. Theory

This approach treats face recognition as a two-dimensional recognition problem, taking advantage of the fact that faces are normally upright and thus may be described by a small set of 2-D characteristic views. Face images are projected onto a feature space ('face space') that best encodes the variation among known face images. The face space is defined by the 'eigenfaces', which are the eigenvectors of the set of faces; they do not necessarily correspond to isolated features such as eyes, ears, and noses. The framework provides the ability to learn to recognize new faces in an unsupervised manner.

## III. Algorithm

- **Training**
  1) Take a set of m images cropped to face
  2) Flatten these images to vectors.
  3) Calculate the average face and subtract from these vectors.
  4) Form a matrix with each normalized face vector as column.
  5) Compute the reduced covariance matrix
  6) Calculate the eigenvectors and eigenvalues of the reduced covariance matrix.
  7) Select k eigenvectors corresponding to k largest eigenvalues. These eigenvectors are called eigenfaces.
  8) Represent each normalized face as linear combination of these eigenfaces.
  9) Store these weights
- **Testing and Recognition**
  1) Given an unknown face centered around the face
  2) Normalize the image
  3) Project the normalized vector into eigenspace to obtain the linear combination of eigenfaces.
  4) Create a error vector by subtracting this weight vector from the training weight vector
  5) Find the image which has the lowest error, if this error is lesser than the threshold, then this is the predicted image.

## IV. observation

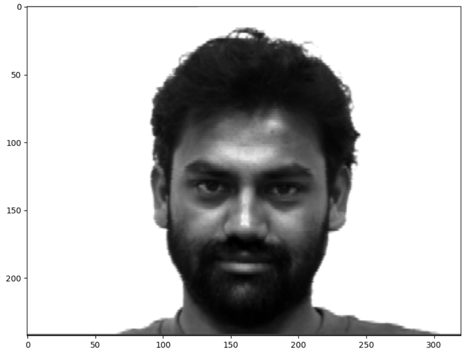

Fig. 1. sample input image

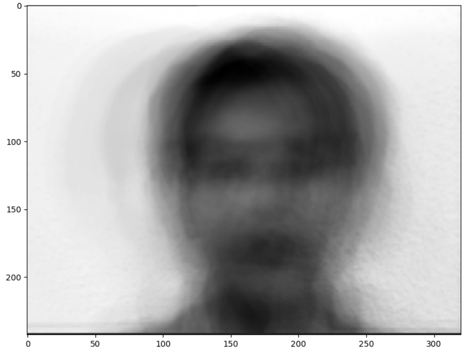

Fig. 2. average face

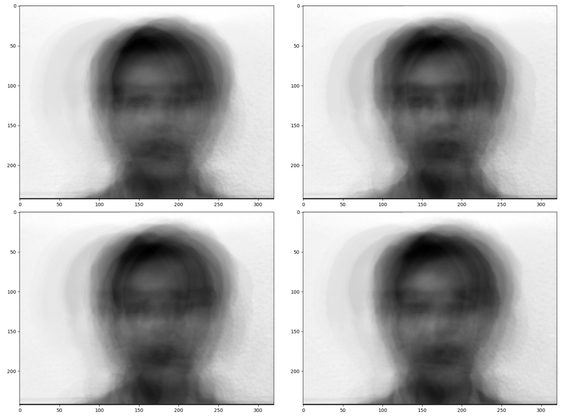

Fig. 3. Four Eigenfaces

## V. Result

- The system was implemeneted in python using numpy and PIL libraries.
- It was then trained on faces from the Yale face database.
- An accuracy of 100% was obtained when tested with a subset from the database which is not included in training.
- Near realtime performance was observed.

| | No. of samples | Accuracy |
|---|---|---|
| Training | 150 | 100% |
| Testing | 15 | 100% |

## VI. Inference

- Face should be properly aligned and should front facing for proper results.