Shri Ramdeobaba College of Engineering and Management
Nagpur, 440013

Department of Computer Science Engineering (AIML)

Deep Learning Lab

Name : *Shantanu Mane*
Roll No. : *E63*
Batch : *CSE-AIML*
Date : *5/3/2023*

AIM - To implement facial reconstruction using PCA, on the faces' dataset.

Importing Dependencies

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

PCA Class

```
class FacialImageCompressionUsingPCA:

    def __init__(self):
        self.facesList = os.listdir("D:\DLLab\Practical 3\data\lfwcrop_grey/faces")
        self.X = []

    def CreateXMatrix(self):
        for face in self.facesList[:10]:
            image = cv.imread(f"D:/DLLab/Practical 3/data/lfwcrop_grey/faces/{face}", 0)
            print(image.shape)
            self.X.append(np.ravel(np.array(image)))

        self.X = np.array(self.X)
        print(self.X)
```

```python
    def GetMeanFace(self):
        meanFace = np.mean(self.X, axis=0)

        plt.imshow(meanFace, cmap='gray')
        plt.show()

    def GetEigenFaces(self):
        XTX = self.X.T.dot(self.X)
        eigenvalues, eigenvectors = np.linalg.eig(XTX)

        print(f"eigenvalues = {eigenvalues}, eigenvectors = {eigenvectors.shape}")

        image = np.zeros((64, 64))
        for i in eigenvectors:
            image += i.reshape(64, 64).dot(self.X[0].reshape(64, 64).T).dot(i.reshape(64, 64)

        plt.imshow(image, cmap='gray')
        plt.show()

    def PCA(self):
        print(self.X[0])
```

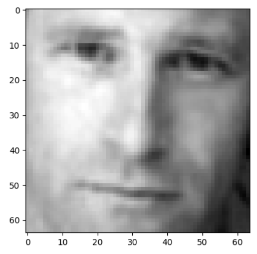
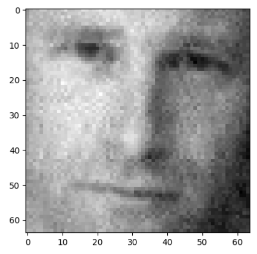
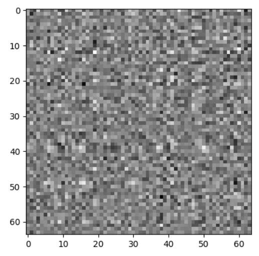
Main Function

```python
from Facial_Image_Reconstruction_Using_PCA import FacialImageCompressionUsingPCA

if __name__ == '__main__':
    fr = FacialImageCompressionUsingPCA()

    fr.CreateXMatrix()
    # fr.GetMeanFace()
    # fr.PCA()
    fr.GetEigenFaces()
```

Output

		
with all eigen-faces	with top eigen-faces	minimal eigen-faces