**Title: Implementation of PCA with ANN algorithm for Face recognition**

Using the face dataset:

Dataset:

<https://github.com/robaita/introduction_to_machine_learning/blob/main/dataset.zip>

Following libraries are used:

Numpy, Scipy for matrix multiplication, finding SVD or Eigen vector etc.

Open CV- Python library for inputting/reading images etc

**Steps for Training**:

1. **Generate the Face Database**:
   * Each face image is represented as a matrix with dimensions m×nm \times nm×n, where x∈mx \in mx∈m and y∈ny \in ny∈n indicate the pixel locations and direction.
   * Assume each face image is a column vector; for ppp images, the size of the face database will be mn×pmn \times pmn×p. Let’s denote the face database as Face\_Dbmn×pFace\\_Db\_{mn \times p}Face\_Dbmn×p​.
2. **Mean Calculation**:
   * Compute the mean of each observation using Mi=∑∑Face\_Db(i,j)j=1pM\_i = \sum \sum Face\\_Db(i,j)\_{j=1}^{p} Mi​=∑∑Face\_Db(i,j)j=1p​ where Mmn×1M\_{mn \times 1}Mmn×1​ represents the mean vector.
3. **Zero Mean Adjustment**:
   * Subtract the mean face from each face image to generate mean-zero face data Δ\DeltaΔ, calculated as Δ(i)mn×p=Face\_Db(i)mn×p−Mmn×1\Delta(i)\_{mn \times p} = Face\\_Db(i)\_{mn \times p} - M\_{mn \times 1}Δ(i)mn×p​=Face\_Db(i)mn×p​−Mmn×1​ for i∈1,2,3,…,pi \in 1,2,3,\dots,pi∈1,2,3,…,p.
4. **Covariance Calculation of Mean-Aligned Faces**:
   * Typically, the covariance is calculated as C=∑(Xi−X)⋅(Yi−Y)TC = \sum (X\_i - X) \cdot (Y\_i - Y)^TC=∑(Xi​−X)⋅(Yi​−Y)T, where XXX and YYY are the means of XiX\_iXi​ and YiY\_iYi​ respectively, and CCC is the covariance matrix.
   * Using this on face data results in Cmn,mn=∑∑∑(Δ(z,i)−Mz,i)⋅(Δ(z,i)−My,i)TC\_{mn,mn} = \sum \sum \sum (\Delta(z,i) - M\_{z,i}) \cdot (\Delta(z,i) - M\_{y,i})^TCmn,mn​=∑∑∑(Δ(z,i)−Mz,i​)⋅(Δ(z,i)−My,i​)T where mnmnmn direction computation becomes complex.
   * Turk and Peterland (1991) proposed a simplified method to compute a surrogate covariance as Cp,p=∑∑∑(Δ(z,i)−Mz,i)⋅(Δ(y,i)−My,i)C\_{p,p} = \sum \sum \sum (\Delta(z,i) - M\_{z,i}) \cdot (\Delta(y,i) - M\_{y,i})Cp,p​=∑∑∑(Δ(z,i)−Mz,i​)⋅(Δ(y,i)−My,i​). This approach focuses on the significant directions (p directions), ignoring the insignificant ones.
5. **Eigenvalue and Eigenvector Decomposition**:
   * Decompose the covariance matrix Cp×pC\_{p \times p}Cp×p​ into eigenvectors Vp×pV\_{p \times p}Vp×p​ and eigenvalues λp×p\lambda\_{p \times p}λp×p​.
6. **Feature Vector Generation**:
   * Select the best directions from ppp by sorting the eigenvalues in descending order. Choose kkk eigenvectors to generate the feature vector Ψp×k\Psi\_{p \times k}Ψp×k​.
7. **Generating Eigenfaces**:
   * Project each mean-aligned face onto the feature vector to generate eigenfaces Φk×mn=Ψk×pT⋅Δp×mnT\Phi\_{k \times mn} = \Psi^T\_{k \times p} \cdot \Delta^T\_{p \times mn}Φk×mn​=Ψk×pT​⋅Δp×mnT​.
8. **Generate Face Signatures**:
   * For each face, project the mean-aligned face onto the eigenfaces to generate a signature ωk×i=Φk×mn⋅Δmn×i\omega\_{k \times i} = \Phi\_{k \times mn} \cdot \Delta\_{mn \times i}ωk×i​=Φk×mn​⋅Δmn×i​ where i∈1,2,3,…,pi \in 1, 2, 3, \dots, pi∈1,2,3,…,p. Here, ω\omegaω will have the dimensions k×pk \times pk×p.
9. **Train the ANN**:
   * Once the best eigenvectors are identified, apply a backpropagation neural network to the data as discussed.

**Steps for Testing**:

1. **Prepare Test Image**:
   * Convert the test image III into a column vector I1mn×1I\_1^{mn \times 1}I1mn×1​.
2. **Zero Mean Adjustment for Testing**:
   * Subtract the mean face MMM from the test face to get I2mn×1=I1mn×1−Mmn×1I\_2^{mn \times 1} = I\_1^{mn \times 1} - M^{mn \times 1}I2mn×1​=I1mn×1​−Mmn×1.

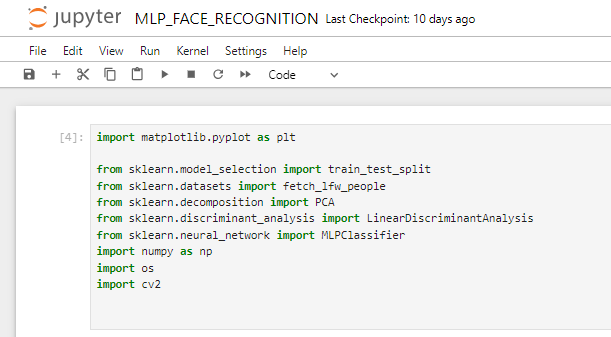
**Projecting Mean-Aligned Face to Eigenfaces**:

* Project the mean-aligned face (I)2(I)\_2(I)2​ onto the eigenfaces (Φ)(\Phi)(Φ) to obtain the projected test face (Ω)(\Omega)(Ω). This can be calculated as (Ω)k×1=(Φ)k×mn×(I)2mn×1(\Omega)\_{k \times 1} = (\Phi)\_{k \times mn} \times (I)\_2^{mn \times 1}(Ω)k×1​=(Φ)k×mn​×(I)2mn×1​.

**Predicting the Unknown Face**:

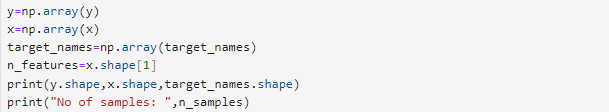
* With the projected test face (Ω)(\Omega)(Ω) and the signature of each face (Φ)(\Phi)(Φ), utilize the trained ANN model to identify the unknown face. Use 60% of the data as the training set and 40% as the test set. Evaluate your classifier based on the following factors:
* **a)** Vary the value of kkk and observe how it impacts classification accuracy. Create a plot showing the relationship between accuracy and different kkk values to illustrate the comparative study.
* **b)** Introduce imposters (individuals not included in the training set) into the test set and recognize them as not being enrolled. Apply the same evaluation.

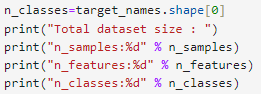
**CODE and OUTPUT:**

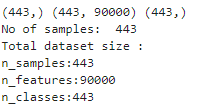
****

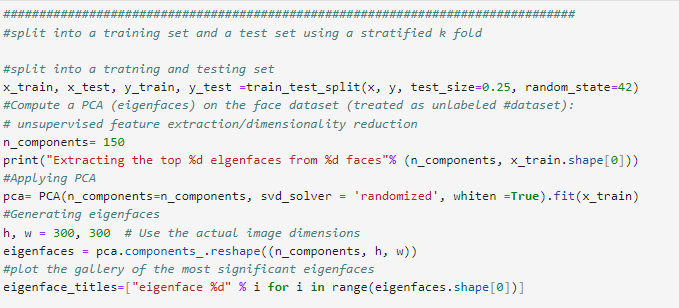
****

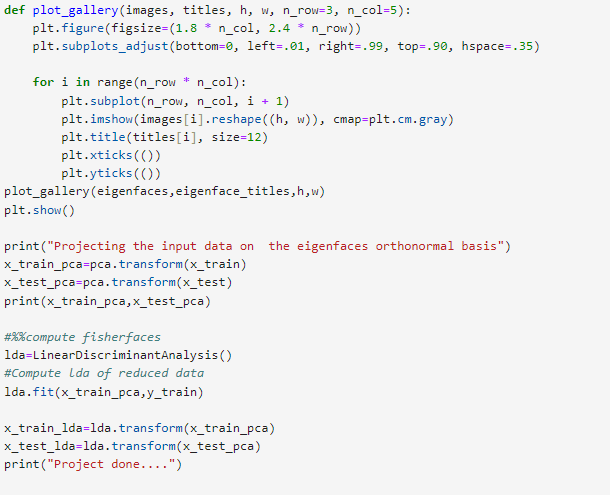
****

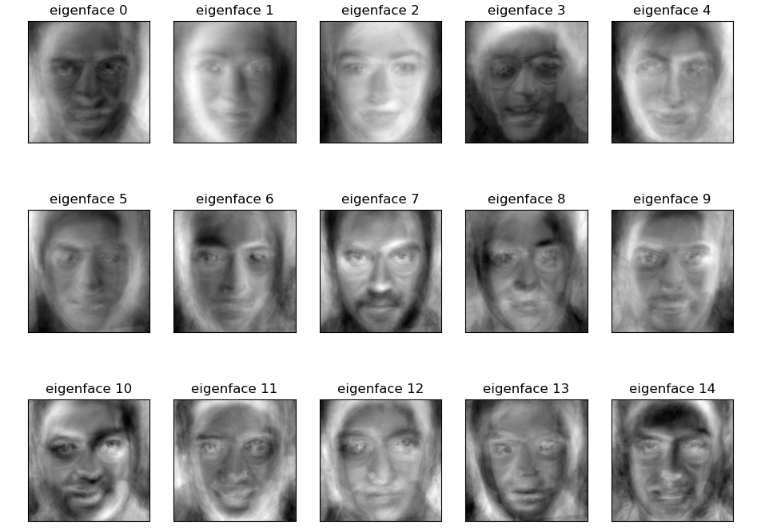
****

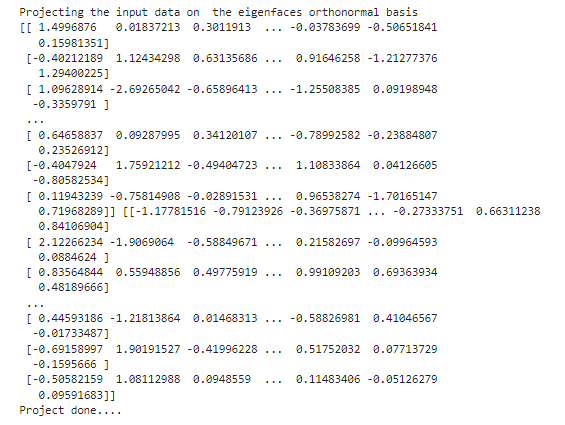
****

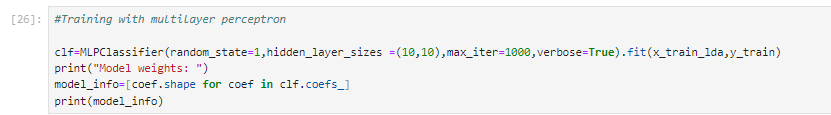
****

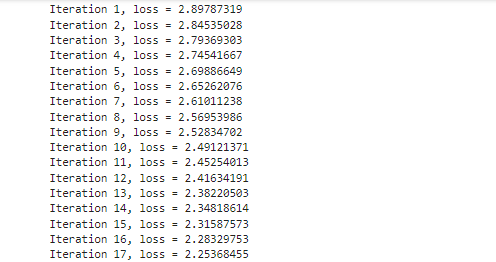
****

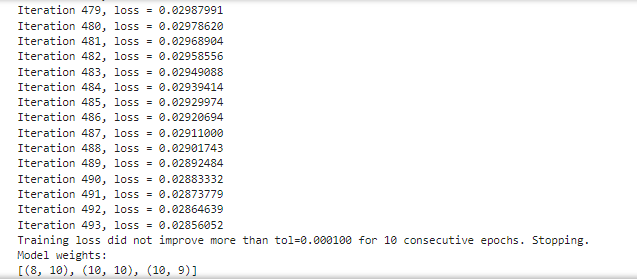
****

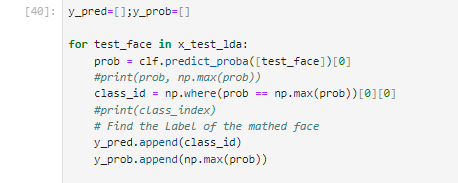
****

****

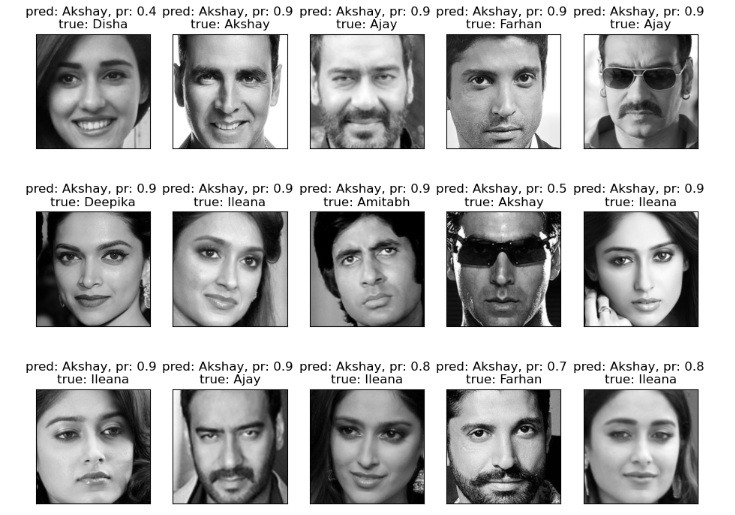
****

****

****

****

****

****