Assignment #1 Face Recognition (Total 150 Points)

Problem Statement

We intend to perform face recognition. Face recognition means that for a given image you can tell the subject id. Our database of subject is very simple. It has 40 subjects. Below we will show the needed steps to achieve the goal of the assignment.

1. Download the Dataset and Understand the Format (10 Points)
    a. ORL dataset is available at the following link. https://www.kaggle.com/kasikrit/att-database-of-faces/downloads/att-database-of-faces.zip/1
    b. The dataset has 10 images per 40 subjects. Every image is a grayscale image of size 92x112.

2. Generate the Data Matrix and the Label vector (10 Points)
    a. Convert every image into a vector of 10304 values corresponding to the image size.
    b. Stack the 400 vectors into a single Data Matrix D and generate the label vector y.
    The labels are integers from 1:40 corresponding to the subject id.

3. Split the Dataset into Training and Test sets (10 Points)
    a. From the Data Matrix D400x10304 keep the odd rows for training and the even rows for testing. This will give you 5 instances per person for training and 5 instances per person for testing.
    b. Split the labels vector accordingly.

4. Classification using PCA (30 points)
    a. Use the pseudo code below for computing the projection matrix U. Define the alpha = {0.8,0.85,0.9,0.95}
    b. Project the training set, and test sets separately using the same projection matrix.

c. Use a simple classifier (first Nearest Neighbor to determine the class labels).
d. Report Accuracy for every value of alpha separately.
e. Can you find a relation between alpha and classification accuracy?

ALGORITHM 7.1. Principal Component Analysis

PCA $(\mathbf{D}, \alpha)$:

1 $\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$ // compute mean
2 $\mathbf{Z} = \mathbf{D} - \mathbf{1}\cdot\boldsymbol{\mu}^T$ // center the data
3 $\boldsymbol{\Sigma} = \frac{1}{n}(\mathbf{Z}^T\mathbf{Z})$ // compute covariance matrix
4 $(\lambda_1, \lambda_2, \ldots, \lambda_d) = \text{eigenvalues}(\boldsymbol{\Sigma})$ // compute eigenvalues
5 $\mathbf{U} = \begin{pmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_d\end{pmatrix} = \text{eigenvectors}(\boldsymbol{\Sigma})$ // compute eigenvectors
6 $f(r) = \frac{\sum_{i=1}^{r}\lambda_i}{\sum_{i=1}^{d}\lambda_i}$, for all $r = 1, 2, \ldots, d$ // fraction of total variance
7 Choose smallest r so that $f(r) \geq \alpha$ // choose dimensionality
8 $\mathbf{U}_r = \begin{pmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_r\end{pmatrix}$ // reduced basis
9 $\mathbf{A} = \{\mathbf{a}_i \mid \mathbf{a}_i = \mathbf{U}_r^T\mathbf{x}_i, \text{for } i = 1, \ldots, n\}$ // reduced dimensionality data

5. Classification Using LDA (30 Points)

a. Use the pseudo code below for LDA. We will modify few lines in pseudocode to handle multiclass LDA.
   i. Calculate the mean vector for every class Mu1, Mu2, ..., Mu40.
   ii. Replace B matrix by S_b.

$$S_b = \sum_{k=1}^{m} n_k(\mu_k - \mu)(\mu_k - \mu)^T$$

Here, m is the number of classes, μ is the overall sample mean, and n_k is the number of samples in the k-th class.

   iii. S matrix remains the same, but it sums S1, S2, S3, ...S40.
   iv. Use 39 dominant eigenvectors instead of just one. You will have a projection matrix $U_{39x10304}$.

b. Project the training set, and test sets separately using the same projection matrix U. You will have 39 dimensions in the new space.
c. Use a simple classifier (first Nearest Neighbor to determine the class labels).
d. Report Accuracy for the Multiclass LDA on the face recognition dataset.
e. Compare the results to PCA results.

ALGORITHM 20.1. Linear Discriminant Analysis

```
LINEARDISCRIMINANT (D = {(x_i, y_i)}_{i=1}^n):
1 D_i ← {x_j | y_j = c_i, j = 1,...,n}, i = 1,2 // class-specific subsets
2 μ_i ← mean(D_i), i = 1,2 // class means
3 B ← (μ_1 − μ_2)(μ_1 − μ_2)^T // between-class scatter matrix
4 Z_i ← D_i − 1_{n_i} μ_i^T, i = 1,2 // center class matrices
5 S_i ← Z_i^T Z_i, i = 1,2 // class scatter matrices
6 S ← S_1 + S_2 // within-class scatter matrix
7 λ_1, w ← eigen(S^{-1} B) // compute dominant eigenvector
```

6. Classifier Tuning (20 Points)

a. Set the number of neighbors in the K-NN classifier to 1,3,5,7.
b. Tie breaking at your preferred strategy.
c. Plot (or tabulate) the performance measure (accuracy) against the K value. This is to be done for PCA and LDA as well.

7. Compare vs Non-Face Images (15 Points)

a. Download non-face images and make them of the same size 92x112. and try to solve the classification problem faces vs. Non-faces.
   i. Show failure and success cases.
   ii. How many dominant eigenvectors will you use for the LDA solution?
   iii. Plot the accuracy vs the number of non-faces images while fixing the number of face images.
   iv. Criticize the accuracy measure for large numbers of non-faces images in the training data.

8. Bonus (5 Points)

a. [5 points] Use different Training and Test splits. Change the number of instances per subject to be 7 and keep 3 instances per subject for testing. compare the results you have with the ones you got earlier with 50% split.

9. Submission Notes

a. Work in groups of 3-4 students.
b. **[25 Points]** You are required to submit a clear and detailed report [in PDF format] illustrating every step in the assignment.

Good Luck

Prof. Dr. Marwan Torki
Eng. Abdelrahman Wael
Eng. Ismail El-Yamany