# BMI-8400 ASSIGNMENT III

**Due Date:** April 27th, 2023 11:59PM

**Points:** 1501

1. Code

The assignment involves implementing and testing a face recognition algorithm using the eigendecompo- sition approach discussed in Lecture 8. For more information on the algorithm, consult the slides and the article posted on Canvas “SlavkovicFaces.pdf”.

# Requirements:

The code should: (1) read in the training images (provided as ASCII pgm files) and store them all in memory; (2) compute the average face and save it as a pgm file; (3) subtract the average face from each column (i.e., from each original image) and store the results in a matrix **A**; (4) compute the eigendecomposition of the covariance matrix of **A**, sorting the eigenfaces by the absolute value of their corresponding eigenvalue; (5) calculate the coordinates of each image in **A** with respect to the subspace spanned by the **top 20 eigenfaces**; (5) for each image in the test set, subtract the average face and compute its coordinates with respect to this subspace; (6) assign each test image to a training image, choosing the training image with the smallest euclidean distance (calculated on the coordinate space).

*•*

Scripting languages such as R, Matlab/Octave, etc. come with functions to perform the eigendecom- position of a matrix. If your language of choice doesn’t come with an in-built function, look for a library/package that does (e.g., in Python you will need to use numpy).

*•*

Parsing the pgm images is very easy, but you can use a package to read in the grayscale values of the images if you wish to do so. However, all the other steps should be implemented by you (except the eigendecomposition of the covariance matrix, of course!).

*•*

1. Dataset

The dataset posted on Canvas under “Files/attDataset” is a modified version of “The Database of Faces”, produced by AT&T Laboratories Cambridge. You will find two folders: (1) train; and (2) test. The train directory contains 240 images of 40 different subjects (identified with the letter “s” followed by an integer in the file name). The test directory contains 160 images of the same subjects, which you will attempt to classify using the nearest neighbor approach described above.

1. Experiments

Using your code, do the following:

1. Save the average face in the training set as a pgm file.
2. Save the top 20 eigenfaces to file.
3. For each image in the test set, return the closest image in coordinate space from the training set.
4. Calculate and report the accuracy of the nearest neighbor classifier as: accuracy = (number of correct predictions) / (total number of predictions).
5. Now modify your algorithm by selecting **20 random eigenfaces**, as opposed to the top 20 eigenfaces you have been using. Report which eigenfaces you randomly selected, and calculate the accuracy of the method using the subspaces spanned by these vectors. You should obtain a lower accuracy value. How do you explain this, given that both subspaces have the same number of dimensions?

1See holistic rubric on grading programming assignments

1

2 BMI-8400 ASSIGNMENT III

1. What to hand in

The assignment should be submitted on Canvas as a single compressed file (either zip or tar), containing the following items:

1. A **source code** folder with all necessary files to run the program and a README file with clear instructions on how to run (and if necessary, compile) the program.
2. A pgm file containing the average training face.
3. A document reporting your answers for the experiments listed above.

# As always, start early and contact the instructor if you need any clarification. You can discuss the assignment and your implementation with classmates, but both the code and the experiments should be your own. If you do discuss the assignment with a classmate, please make a note about it in the report.