CV (Spring 2021)

Name:

Project 05. Data Reduction with Eigenfaces of the *CS department at UTRGV*

**This project is worth 200 points.**

**Goal:** Your goal in this project is to create a Python program to gain deep understanding of PCA as a data reduction and representation technique. You will:

1. Collect the images to be used from the CS dept. official website.
2. Center and normalize the data yourself using NumPy operations.
3. Create the matrix of the normalized data.
4. Calculate the co-variance matrix.
5. Calculate the eigen values and eigen vectors.
   1. Then decide on a subset of them to keep based on %variance explained.
6. Create the transform/projection matrix for each image.
   1. Your stored data would be composed off two parts:
      1. The mean and eigenfaces (common to all images in the data set)
      2. Projection coefficients of each image (each set of coefficients is specific to the image)

The data reduction comes from two sources: there only one set of eigenfaces to be stored; the images coefficients take only a tiny amount of storage

1. Represent an image in terms of the kept eigenvectors.

You will represent your data (images) using a subset of the computed PCA components of the images data set. Once the PCAs are computed based on the co-variance matrix, you will find how many PCA components you will need to account for a given percentage of the variance. You will try 75%, 80%, 85%, and 90%. Select 4 images and represent them in terms of the components, showing in the process how the image representation improves as you add more components.

Your program must show the following:

1. The mean image of the data set
2. The top 10 PCA eigenvectors (eigenfaces with largest eigenvalues) as images
3. The representation of at least 2 images using these 10 components cumulatively:
   1. Mean, mean+1stPCA component, mean+2ndPCA component, …mean+10thPCA component.

**Data Set:**

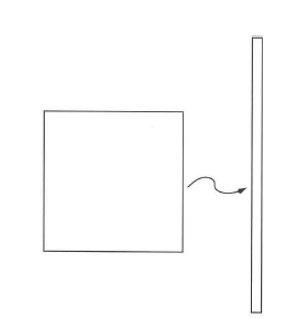
Your data set will consist of the pictures of the CS department to be taken from our official website.

**Image Preprocessing:**

PCA input images need to be carefully preprocessed. You will preprocess your images in a similar way to what is shown below. Images will be in gray-levels and restricted to the frontal face area—ear to ear and chin to top of forehead. Images are then to be resized to 125x150 (width x height). Each image is then converted into a vector by stacking the rows.

****

Each image is then converted into a vector by stacking the rows:



**For this part, you are allowed to work with other students to split the preprocessing work.**

**Mean Image:**

When the mean image is calculated, it should look something like this:

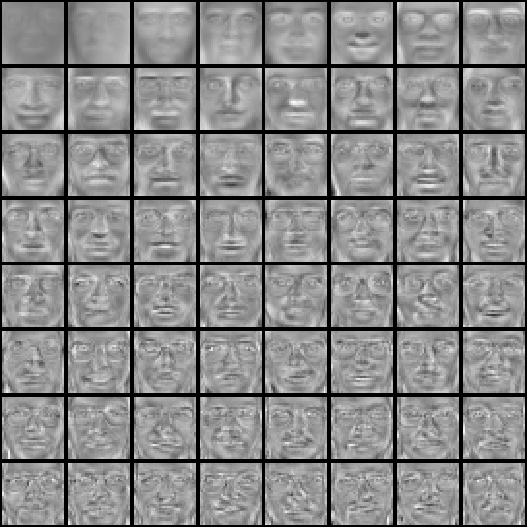


**Top eigenvector Image:**

The top eigenvectors: u1,…uk (visualized as images - eigenfaces) should look something like this:

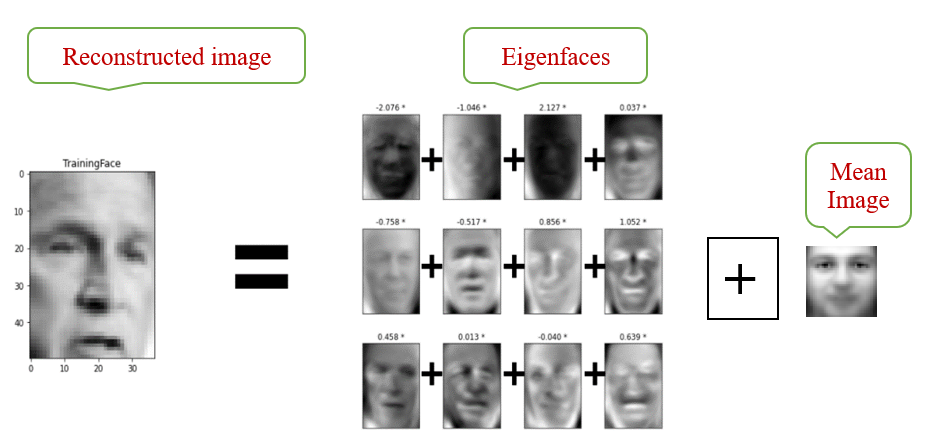
**u2**

**u1**

****

**Image Representation:**

Any image, **x**, is then represented as a linear combination of the mean and the above eigenfaces. The coefficients of each eigenvector image depend on **x** (exactly on the projection of the **x** on that specific eigenvector). The following figure illustrates the linear combinations needed to reconstruct George Bush’s image from the first 12 PCAs. The number above each eigenface is the projection specific to Bush’s image and would be different for each image in the set.

****

**Storage Reduction:**

Compute the amount of storage (in Bytes) savings achieve using PCA. The formula could be derived as the difference between the following two quantities:

**Original storage = M \* (size of each gray level image after preprocessing)**

**New storage =** Size of Mean Image **+**

Size of eigenfaces kept (K \* size of each Eigenface) **+**

Size of coefficients for all the images (M \* (K\* size of one coefficient)).

Check this reference for the steps in more detail (same as in class notes). We are only doing the representation not the detection or the classification.

[**https://www.geeksforgeeks.org/ml-face-recognition-using-eigenfaces-pca-algorithm/**](https://www.geeksforgeeks.org/ml-face-recognition-using-eigenfaces-pca-algorithm/)

**Grading and Submission Guide:**

* Must submit the whole project (python folder with code, image dataset, and results) zipped using 7zip tools with the name: LastName\_FirstName\_Project-05.
* You report should go through the above steps in detail and show resulting images at each stage
* This is an **individual** project: The work should represent your own: that you acknowledge that have not incorporated into this project any unacknowledged material from the work of another person, including papers, words, ideas, information, computer code, data, evidence-organizing principles, or style of presentation taken from the Internet, books, periodicals, or other sources.