ISYE 6740 Spring 2025
Homework 2
(100 points + 5 bonus points)

Provided Data:

- Q2: PCA: Food Consumption [food-consumption.csv]
- Q3: Order of faces using ISOMAP [isomap.dat, isomap.mat]
- Q5: Eigenfaces and simple face recognition [yalefaces]

All results that are present only in code/notebooks, but not in your PDF report will not be accepted for points.
For any code that requires randomness, please set your seed as 6740

1. Conceptual questions [30 points].

1. (5 points) Please prove the first principle component direction v corresponds to the largest eigenvector of the sample covariance matrix:

$$v = \arg \max_{w:\|w\|\leq 1} \frac{1}{m} \sum_{i=1}^{m} (w^T x^i - w^T \mu)^2$$

You may use the proof steps in the lecture, but please write them logically and cohesively.

Rewriting the given expression while representing mean-centered data ($\bar{x}^i = x^i - \mu$):

$$\frac{1}{m} \sum_{i=1}^{m} \left(w^T x^i - w^T \mu\right)^2 = \frac{1}{m} \sum_{i=1}^{m} (w^T \bar{x}^i)^2 = w^T \left(\frac{1}{m} \sum_{i=1}^{m} \bar{x}^i (\bar{x}^i)^T \right) w = w^T C w,$$

where $\left(\frac{1}{m} \sum_{i=1}^{m} \bar{x}^i (\bar{x}^i)^T\right) = C$, the sample covariance matrix.

The goal can now be written as:

$$v = \max_{w:\|w\|=1} w^T C w$$

The Langrangian method can be used to solve this:

$$L(w, \lambda) = w^T C w - \lambda (w^T w - 1)$$
$$\frac{\partial L}{\partial w} = 2Cw - 2\lambda w$$
$$Cw = \lambda w$$

This shows that w must be an eigenvector of C, and λ is the corresponding eigenvalue. The value $w^T C w$ is maximized when w aligns with the eigenvector corresponding to the largest eigenvalue of C. Thus, the first principal component direction is the eigenvector of C associated with its largest eigenvalue.

2. (5 points) Based on your answer to the question above, explain how to further find the second and third largest principle component directions.

To find the second principal component direction:

(a) Must be orthogonal to v_1 ($v_2^T v_1 = 0$)

(b) Maximizes the variance $v_2^T C v_2$ subject to these orthogonality constraints and $\|v_2\| = 1$

(c) Corresponds to the eigenvector of C associated with the second largest eigenvalue λ_2

To find the third principal component direction:

(a) Must be orthogonal to both v_1 and v_2 ($v_3^T v_1 = 0$ and $v_3^T v_2 = 0$)

(b) Maximizes the variance $v_3^T C v_3$ subject to these orthogonality constraints and $\|v_3\| = 1$

(c) Corresponds to the eigenvector of C associated with the third largest eigenvalue λ_3

The general process to find these principle component directions is:

(a) Compute the eigendecomposition of the covariance matrix

(b) Assign the eigenvectors in decreasing order of their eigenvalues to principal component directions

3. (5 points) Consider the diagonal matrix $A = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}$. Find two distinct eigenvalue decompositions of A, and prove mathematically that both are valid. This demonstrates that eigendecomposition is not unique. **Note:** Your eigenvalue decompositions must be done mathematically, not programatically.

For $A = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}$:

(a) Eigenvalues: $\lambda_1 = 3, \lambda_2 = 2$

(b) Eigenvectors: $q_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, q_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Combining the eigenvectors as columns, a potential $Q_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ gives the following eigenvalue decomposition:

$$A = Q_1 \Lambda Q_1^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}$$

Another potential $Q_2 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$ comes from negating q_1 as the first column, and gives the following eigenvalue decomposition:

$$A = Q_2 \Lambda Q_2^T = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -3 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}$$

Both decompositions give $A = Q\Lambda Q^T$, but the eigenvectors differ in sign:

(a) In the first decomposition, $Q_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

(b) In the second decomposition, $Q_2 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$

The eigenvalue decomposition is not unique because the eigenvectors of a symmetric matrix can be scaled while still satisfying the decomposition.

4. (5 points) Explain the three key ideas in ISOMAP (for manifold learning and non-linear dimensionality reduction).

(a) For each data point x^i, identify its neighbors within a threshold ϵ and create the adjacency matrix A, where each entry represents the Euclidean distance between connected points

(b) Using the graph defined by A, compute the shortest path distances between all pairs of points x^i and x^j, ensuring the distances account for the non-linear manifold structure, by using algorithms like Floyd-Warshall or Dijkstra's

(c) Use Multidimensional Scaling on the distance matrix D to embed the data in a low-dimensional space while preserving the distance relationships in D

This preserves the "walking distance" on the manifold, capturing its intrinsic geometry efficiently.

5. (5 points) Explain how to decide k, the number of principle components, from data.

To decide the number of principal components (k) in PCA, we can look for how much variance each component explains. A k is chosen so the total variance captured is at least a certain threshold (e.g., 95%), or use a scree plot to find the elbow point.

6. (5 points) How do outliers affect the performance of PCA? You can create numerical examples to study and show this.

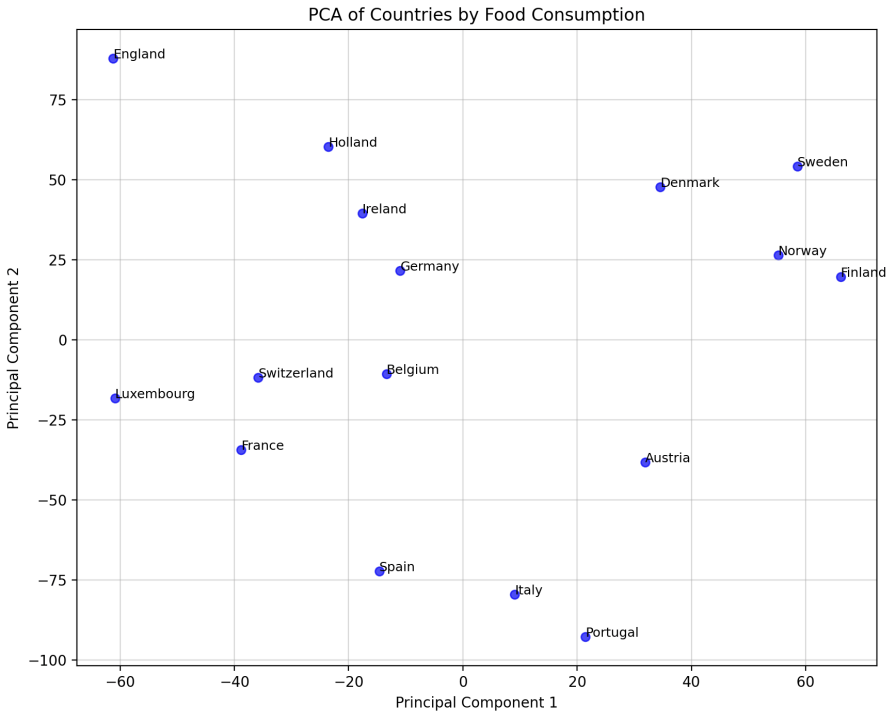
Outliers can distort PCA results by skewing the covariance matrix and pulling principal components toward extreme values. To address this, it is common to remove outliers before applying PCA. Robust PCA methods, which are designed to handle outliers, treat the problem as a convex optimization challenge to separate the low-rank structure of the data from sparse noise caused by outliers. This approach ensures that the principal components reflect the underlying data structure rather than being dominated by a few extreme points, as described by Candes, Li, Ma, and Wright (2009).

2. PCA: Food consumption in European countries [20 points].

The data food-consumption.csv contains 16 countries in Europe and their consumption for 20 food items, such as tea, jam, coffee, yogurt, and others. We will perform principal component analysis to explore the data. In this question, please implement PCA by writing your own code (you can use any basic packages, such as numerical linear algebra, reading data, in your file).

First, we will perform PCA analysis on the data by treating each country's food consumption as their "feature" vectors. In other words, we will find weight vectors to combine 20 food-item consumptions for each country.

1. (10 points) For this problem of performing PCA on countries by treating each country's food consumption as their "feature" vectors, explain how the data matrix is set-up in this case (e.g., the columns and the rows of the matrix correspond to what). Now extract the first two principal components for each data point (thus, this means we will represent each data point using a two-dimensional vector). Draw a scatter plot of two-dimensional representations of the countries using their two principal components. Mark the countries on the plot (you can do this by hand if you want). Please explain any pattern you observe in the scatter plot.

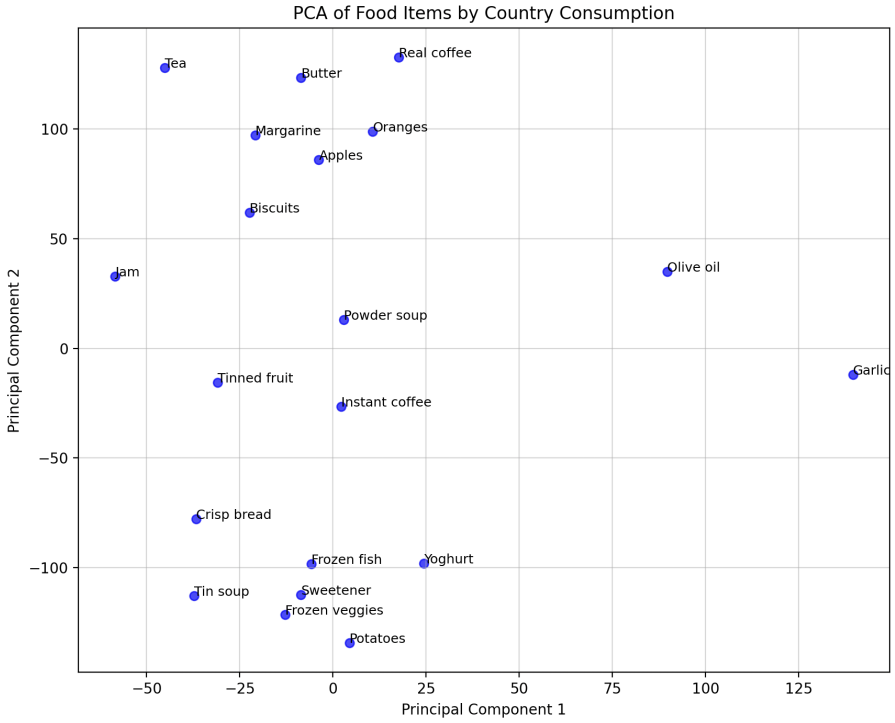


When performing PCA with countries as feature vectors, the data matrix is set up so that rows represent food items, and columns represent countries. Each value reflects the consumption of a specific food item in a specific country. By projecting the data onto the first two principal components, we can capture the main patterns in how food consumption varies across countries.

In the scatter plot, countries that have similar dietary habits are grouped closer together. For example, Sweden, Norway, and Finland form a cluster, likely due to shared regional preferences. On

the other hand, Italy, Spain, and Portugal are positioned further apart, reflecting their Mediterranean diets, which include more olive oil and fresh produce. The clear separation between regions highlights how PCA can effectively group countries with similar food consumption behaviors.

2. (10 points) Now, we will perform PCA analysis on the data by treating country consumptions as "feature" vectors for each food item. In other words, we will now find weight vectors to combine country consumptions for each food item to perform PCA another way. Project data to obtain their two principle components (thus, again each data point – for each food item – can be represented using a two-dimensional vector). Draw a scatter plot of food items. Mark the food items on the plot (you can do this by hand if you want). Please explain any pattern you observe in the scatter plot.



For PCA with food items as feature vectors, the data matrix is organized so that rows represent countries, and columns represent food items. Each value shows the amount of a specific food item consumed in a particular country. By projecting the data onto two principal components, we can see patterns in how food items are consumed across Europe.

The scatter plot reveals some interesting groupings. For instance, instant coffee and powder soup are positioned near each other, indicating they are often consumed in similar countries or contexts. Meanwhile, olive oil and garlic stand out on the plot, reflecting their strong association with Mediterranean cuisines. These patterns show how PCA can group food items based on shared consumption trends, giving insights into regional dietary preferences.

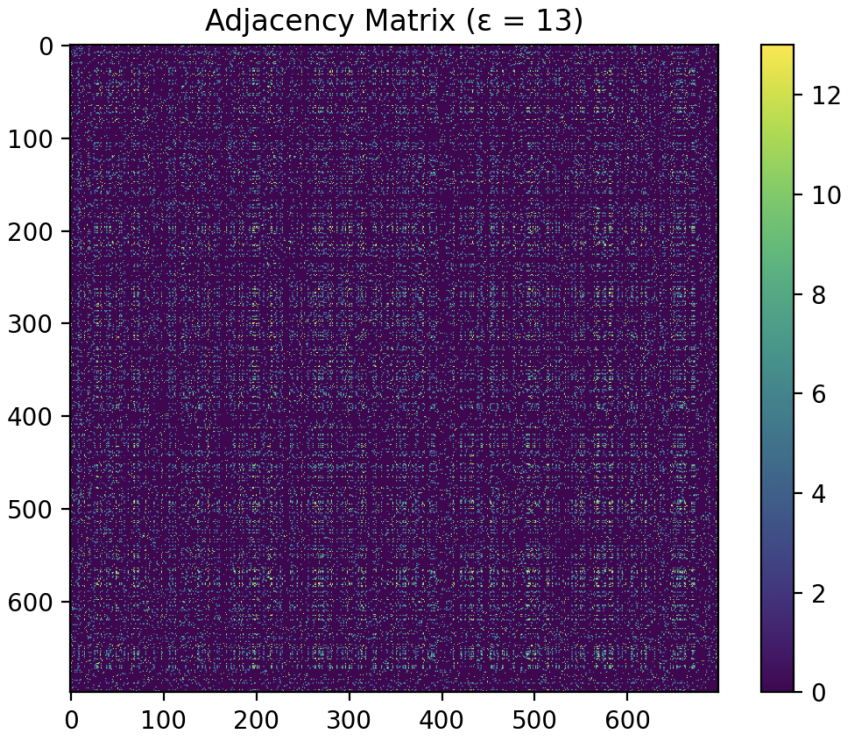
3. Order of faces using ISOMAP [25 points]

This question aims to reproduce the ISOMAP algorithm results in the original paper for ISOMAP, J.B. Tenenbaum, V. de Silva, and J.C. Langford, Science 290 (2000) 2319-2323 that we have also seen in the lecture as an exercise (isn't this exciting to go through the process of generating results for a high-impact research paper!)

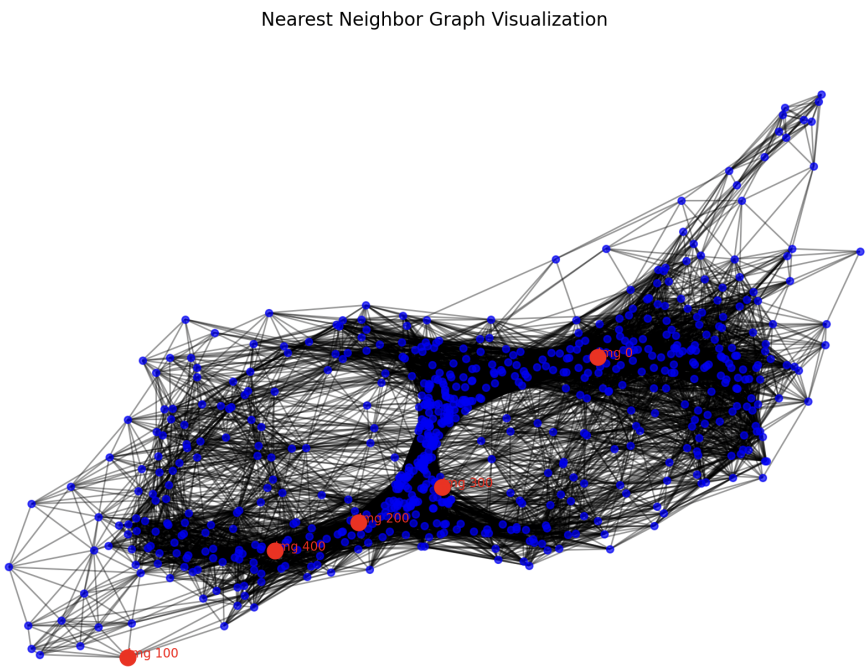
The file isomap.mat (or isomap.dat) contains 698 images, corresponding to different poses of the same face. Each image is given as a 64×64 luminosity map, hence represented as a vector in $\mathbb{R}^{4096}$. This vector is stored as a row in the file. (This is one of the datasets used in the original paper.) In this question, you are expected to implement the ISOMAP algorithm by coding it up yourself. You may find the shortest path (required by one step of the algorithm), using `https://docs.scipy.org/doc/scipy/reference/generated/scipy.sparse.csgraph.shortest_path.html`.

Using Euclidean distance (i.e., in this case, a distance in $\mathbb{R}^{4096}$) to construct the ϵ-ISOMAP (follow the instructions in the slides.) You will tune the ϵ parameter to achieve the most reasonable performance. Please note that this is different from K-ISOMAP, where each node has exactly K nearest neighbors.

1. (5 points) Visualize the nearest neighbor graph (you can either show the adjacency matrix (e.g., as an image), or visualize the graph similar to the lecture slides using graph visualization packages such as Gephi (https://gephi.org) and illustrate a few images corresponds to nodes at different parts of the graph, e.g., mark them by hand or use software packages).



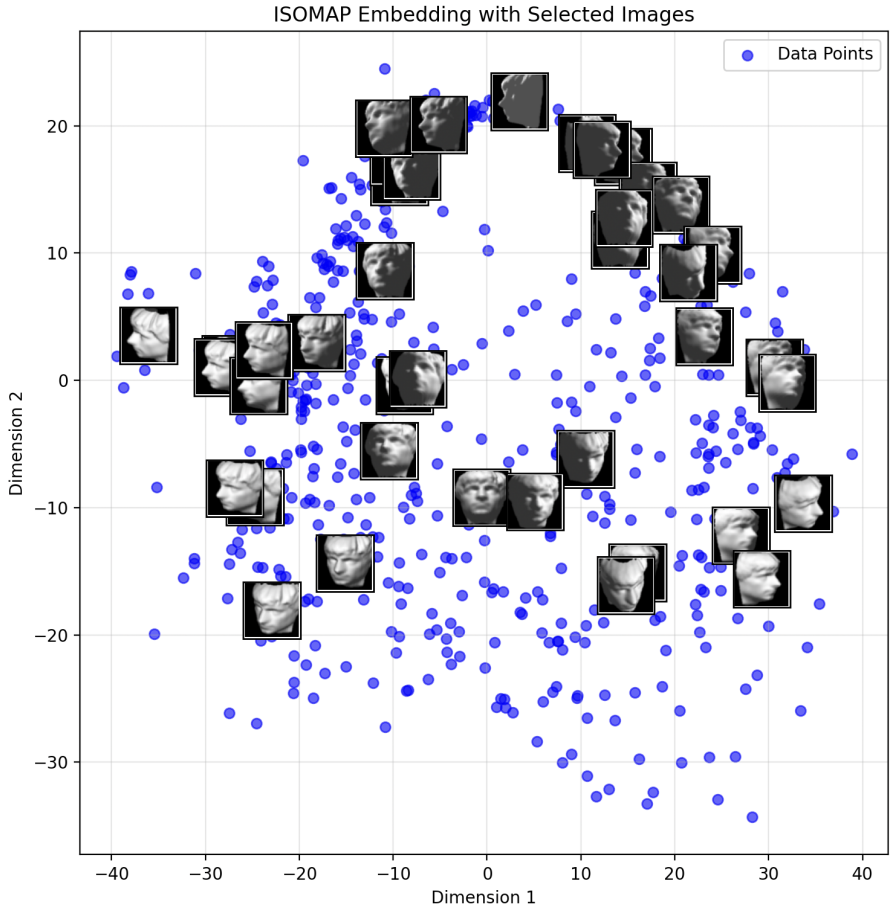
The adjacency matrix visualization represents the connectivity between data points in the ISOMAP algorithm based on a chosen epsilon value. Each cell in the matrix corresponds to the distance between two images (nodes) in the dataset, with connections drawn only if their distance falls below epsilon. The matrix is symmetric, as distances between pairs of nodes are bidirectional. Brighter cells in the adjacency matrix indicate shorter distances, showing closely connected nodes.



In the nearest neighbor graph visualization, each node represents an image, and edges connect nodes within the defined epsilon range. Specific images (highlighted in red) are marked as key nodes to illustrate their positions in the graph relative to others. The graph structure reveals clusters and patterns, such as nodes grouped based on pose or orientation similarities in the image dataset. These clusters align with how ISOMAP captures the underlying low-dimensional manifold structure of the data.

This step helps visualize relationships and ensures the epsilon value is appropriately chosen to capture meaningful connections without making the graph overly dense or sparse.

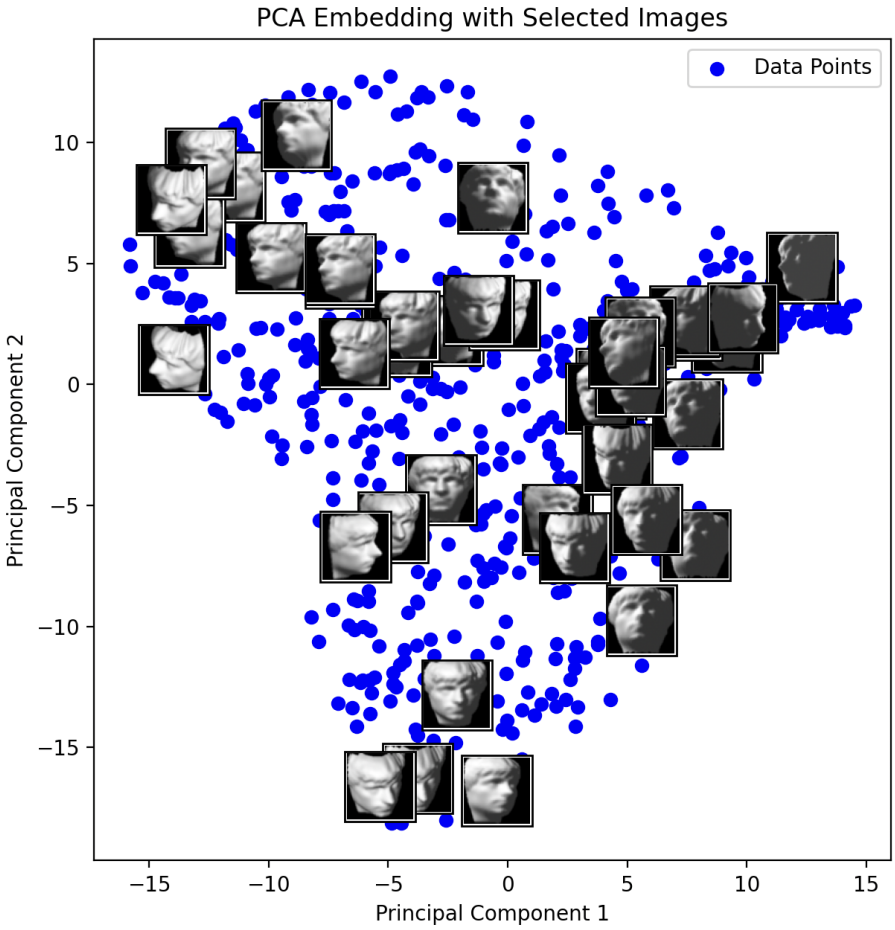
2. (10 points) Implement the ISOMAP algorithm yourself to obtain a two-dimensional low-dimensional embedding. Plot the embeddings using a scatter plot, similar to the plots in lecture slides. Find a few images in the embedding space and show what these images look like and specify the face locations on the scatter plot. Comment on do you see any visual similarity among them and their arrangement, similar to what you seen in the paper? Come up with a way to tune the kernel bandwidth to have desired result.



The ISOMAP embedding reveals clear clusters of faces based on their orientation, with similar poses grouped together and gradual transitions forming a circular pattern. This reflects ISOMAP's ability to capture nonlinear relationships, consistent with findings from the original paper. Faces with similar orientations appear closer in the embedding, while those with distinct poses are further apart.

Tuning the kernel bandwidth is key to achieving a meaningful embedding. A small epsilon can create disconnected clusters, while a large epsilon may smooth out important details. The approach I used is testing different epsilon values, ensuring the graph is connected but not overly dense, and validating the embedding for clear clusters and transitions. This balance preserves both local and global structures, producing an accurate representation that is interpretable.

3. (10 points) Perform PCA (you can now use your implementation written in Question 1) on the images and project them into the top 2 principal components. Again show them on a scatter plot. Explain whether or you see a more meaningful projection using ISOMAP than PCA.



The comparison between PCA and ISOMAP showcases ISOMAP's ability to better capture the nonlinear relationships in the data. While PCA organizes images based on variance, its projection lacks meaningful clustering of similar poses or orientations. ISOMAP, however, preserves the global structure, grouping faces with similar angles and orientations more intuitively. This makes the ISOMAP mapping more effective for visualizing the complex structure of the facial pose data.

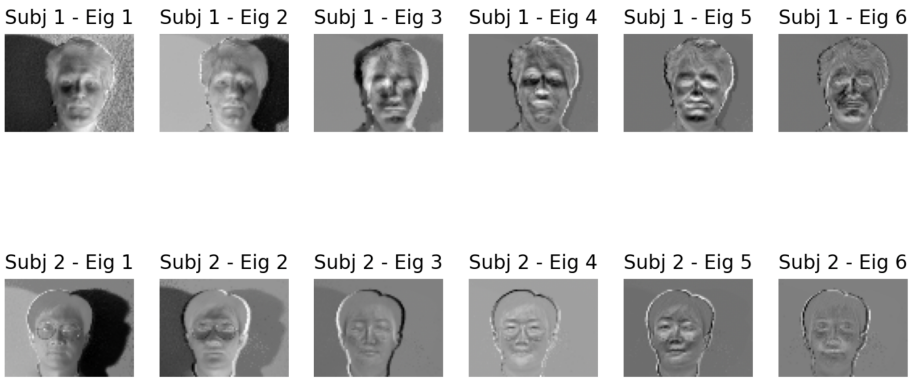
4. Eigenfaces and simple face recognition [25 points].

This question is a simplified illustration of using PCA for face recognition. We will use a subset of data from the famous Yale Face dataset.

Remark: You will have to perform downsampling of the image by a factor of 4 to turn them into a lower resolution image as a preprocessing (e.g., reduce a picture of size 16-by-16 to 4-by-4). In this question, you can implement your own code or call packages.

First, given a set of images for each person, we generate the eigenface using these images. You will treat one picture from the same person as one data point for that person. Note that you will first vectorize each image, which was originally a matrix. Thus, the data matrix (for each person) is a matrix; each row is a vectorized picture. You will find weight vectors to combine the pictures to extract different "eigenfaces" that correspond to that person's pictures' first few principal components.

1. (10 points) Perform analysis on the Yale face dataset for Subject 1 and Subject 2, respectively, using all the images EXCEPT for the two pictures named subject01-test.gif and subject02-test.gif. **Plot the first 6 eigenfaces for each subject.** When visualizing, please reshape the eigenvectors into proper images. Please explain can you see any patterns in the top 6 eigenfaces?



(a) i. Subject 1: The eigenfaces strongly emphasize overall facial structure and edges, highlighting prominent features like cheekbones, forehead, and jawline

ii. Subject 2: There is a clear focus on glasses and facial contours in the eigenfaces, which biases the representation of this subject toward eyewear

(b) The first two eigenfaces for both subjects primarily capture shadows and lighting variations across the faces. These components represent the largest variance in the dataset, as lighting significantly impacts the pixel intensity distribution.

(c) While both subjects have some images with glasses in the dataset, Subject 2's eigenfaces exhibit a stronger bias toward glasses. This could lead to issues in recognition. If a test image of Subject 1 with glasses is presented, it might be misclassified as Subject 2 because of the overrepresentation of eyewear features in Subject 2's eigenfaces. Similarly, a test image of Subject 2 without glasses may not match the dominant features of Subject 2's eigenfaces, potentially leading to misclassification.

2. (10 points) Now we will perform a simple face recognition task.

   Face recognition through PCA is proceeded as follows. Given the test image subject01-test.gif and subject02-test.gif, first downsize by a factor of 4 (as before), and vectorize each image. Take the top eigenfaces of Subject 1 and Subject 2, respectively. Then we calculate the *projection residual* of the 2 vectorized test images with the vectorized eigenfaces:

$$s_{ij} = \|(\text{test image})_j - (\text{eigenface}_i)(\text{eigenface})_i^T(\text{test image})_j\|_2^2$$

   Report all four scores: s_{ij}, $i = 1, 2$, $j = 1, 2$. Explain how to recognize the faces of the test images using these scores.

   (a) Subject 1 Test Image 1 (s11): 42756731.412235096

   (b) Subject 1 Test Image 2 (s12): 253675711.18683523

   (c) Subject 2 Test Image 1 (s21): 275194826.0725945

   (d) Subject 2 Test Image 2 (s22): 28211094.58302682

   A projection residual measures how well the test image can be reconstructed using the eigenfaces of a particular subject. Specifically, for each eigenface, the projection of the test image is calculated by projecting the test image onto the eigenface and then subtracting this projection from the test image itself. The squared norm of this difference is computed to quantify the residual for each eigenface, and these residuals are summed across all eigenfaces for a given subject.

   The classification decision is based on comparing these residuals. The idea is that a test image will have the smallest residual when projected onto the eigenfaces of the correct subject because these eigenfaces best capture the variance and features specific to that subject.

   If $s_{11} < s_{21}$, Test Image 1 is classified as belonging to Subject 1 because the eigenfaces of Subject 1 better represent it than those of Subject 2. Similarly, if $s_{22} < s_{12}$, Test Image 2 is classified as belonging to Subject 2. This logic ensures that the classification is driven by the reconstruction accuracy of the eigenfaces for each subject, using the fact that eigenfaces capture the most distinctive features of each subject's face.

3. (5 points) Comment if your face recognition algorithm works well and discuss how you would like to improve it if possible.

   This algorithm works reasonably well, as it correctly classifies the test images based on projection residuals. However, there are areas for improvement. The algorithm shows sensitivity to biases in the training data, such as the overrepresentation of glasses in Subject 2's eigenfaces, which could lead to misclassification in edge cases. Additionally, the algorithm is heavily influenced by lighting and shadow variations, as seen in the first two eigenfaces of both subjects. To improve, preprocessing the images with normalization to reduce the impact of lighting and including a more balanced dataset to capture variations in features like glasses would be effective.

5. To subtract or not to subtract, that is the question [Bonus: 5 points].

In PCA, we have to subtract the mean to form the covariance matrix

$$C = \frac{1}{m}\sum_{i=1}^{m}(x^i - \mu)(x^i - \mu)^T$$

before finding the weight vectors, where $\mu = \frac{1}{m}\sum_{i=1}^{m} x^i$. For instance, we let

$$Cw^1 = \lambda_1 w^1$$

where λ_1 is the largest eigenvalue of C, and w^1 is the corresponding largest eigenvector.

Now suppose Prof. X insisting not subtracting the mean, and uses the eigenvectors of

$$\tilde{C} = \frac{1}{m}\sum_{i=1}^{m} x^i x^{iT}$$

to form the weight vectors. For instance, she lets $\tilde{w}^1$ to be such that

$$\tilde{C}\tilde{w}^1 = \tilde{\lambda}_1 \tilde{w}^1$$

where $\tilde{\lambda}_1$ is the largest eigenvalue of $\tilde{C}$.

Now the question is, are they the same (with and without subtract the mean)? Is w^1 equal or not equal to $\tilde{w}^1$? Use mathematical argument to justify your answer.

$$C = \frac{1}{m}\sum_{i=1}^{m} x^i x^{iT} - \mu\mu^T$$

$$\tilde{C} = \frac{1}{m}\sum_{i=1}^{m} x^i x^{iT}$$

$$C = \tilde{C} - \mu\mu^T$$

$$Cw^1 = \lambda_1 w^1 \quad \text{and} \quad \tilde{C}\tilde{w}^1 = \tilde{\lambda}_1 \tilde{w}^1$$

Substituting $C = \tilde{C} - \mu\mu^T$ into the first equation:

$$(\tilde{C} - \mu\mu^T)w^1 = \lambda_1 w^1$$

$$\tilde{C}w^1 = \lambda_1 w^1 + \mu\mu^T w^1$$

This equation shows that w^1 is generally not an eigenvector of $\tilde{C}$, because the term $\mu\mu^T w^1$ introduces a bias that depends on the mean vector. Therefore, $w^1 \neq \tilde{w}^1$ in general.