ISYE 6740 Homework 2

Nick DiNapoli, ndinapoli6@gatech.edu

Due: September 19, 2021

1 Conception questions

1.1 Please explain why the first principle component direction (the weight vector) corresponds to the largest eigenvector of the sample covariance matrix.

We aim to find a directions or weight vectors (w^is) such that the variance of the data along w is maximized,

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

which, through some linear algebra, reduces to the contrained optimization problem of maximizing,

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

or,

$$w^T C w \tag{3}$$

still subject to $||w|| \leq 1$. After forming the Lagrangian function and taking the derivative with respect to w, optimization problem reduces further to,

$$\begin{aligned} Cw &= \lambda w \\ w^T C w &= \lambda ||w||^2 \end{aligned} \tag{4}$$

an eigenvector-eignevalue problem. So the directions $w^1, w^2, ...$ have the largest variances and are eigenvectors of C with corresponding largest eigenvalues $\lambda^1, \lambda^2, ...$ that are the largest variances.

1.2 What is the relationship between SVD and eigendecomposition?

SVD deals with real matricies (i.e. M) that aren't necessarily square (n x m) where n $\leq$ m. M is composed of left singular vectors, singular vectors, and right singular vectors (where the left and right are orthonormal),

$$M = U\Sigma V^T \tag{5}$$

A pair of singular vectors satisfies,

$$\begin{aligned} Mv &= \sigma u \\ M^T u &= \sigma v \end{aligned} \tag{6}$$

so if C = MM^T,

$$\begin{aligned} C &= U\Sigma V^T V \Sigma^T U^T \\ C &= U\Sigma\Sigma^T U^T \end{aligned} \tag{7}$$

and U are the eigenvectors of C with eigenvalues, σ_i^2 (taking C as the covariance matrix).

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

ISOMAP is a nonlinear dimensionality reduction technique with many key ideas. When there is nonlinear structure in the data, the Euclidean distance is not a good distance measure globally. For long range distances in a data cloud, the geodesic distance should be used. The key ideas of ISOMAP include, producing a low dimensional representation which preserves the walking distance over the manifold. The next idea is to find neighbors of each data point, x^i, within some distance, ϵ, and record the neighbors' Euclidean distance in an adjacency matrix. Using this adjacency matrix, the shortest path distance matrix can be obtained by computing the geodesic distance between each pair of points, x^i and x^j. Lastly, with the use of a centering matrix, an eignedecomposition problem can be formed and solved to compute a new low dimensional representation which preserves the information in the distance matrix. This helps to "unfold" the manifold to find intrinsic structures.

2 Eigenfaces and simple face recognition

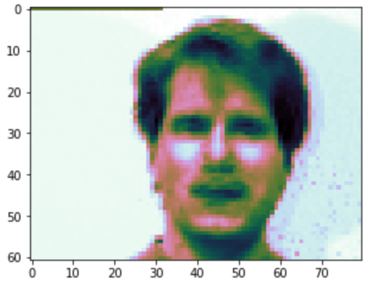
In this problem, I aim to use PCA for face recognition using a subset of the Yale Face dataset. I use two subjects (people) and construct two different data matricies where each data point (row) is a vectorized, 4x downsampled image of that same person. The images include but are not limited to the person with glasses, smiling, sad, sleepy, surprised, winking, etc. I find weight vectors via my own implementation of PCA to combine the images to discover different "eigenfaces" which correspond to each subject's first few principal components (PC).

2.1 PCA and eigenfaces of subjects 1 & 2

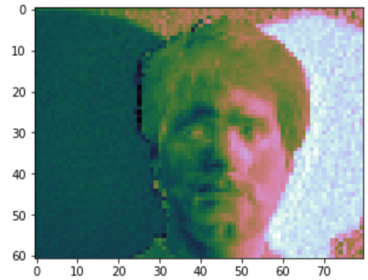
I first applied the preprocessing steps according to the above introduction to all of the images for both subjects except for the "test" images which will be used in the next section. Subject 1 has ten, 243 x 320 images and subject 2 has nine. The test images (straight faces) of each subject are shown in Figure 1 as reference. After downsampling, each image gets reduced to 61 x 80. After my implementation of PCA, I can reshape to visualize the first 6 eigenfaces of each subject. Figures 2-5 show these visualizations.

(a) Subject 1 (243 x 320) (b) Subject 2 (243 x 320)

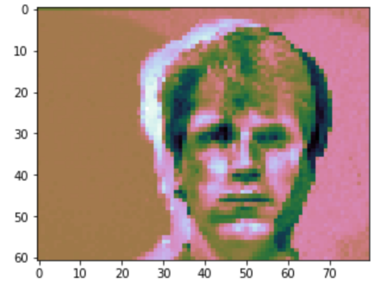
Figure 1: Test images for subjects 1 and 2



(a) Eigenface 1

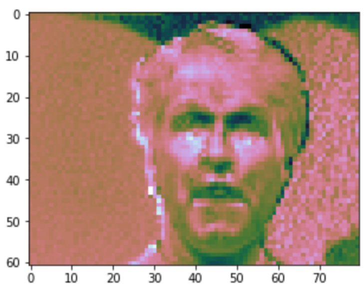


(b) Eigenface 2

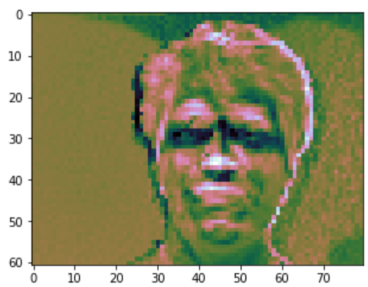


(c) Eigenface 3

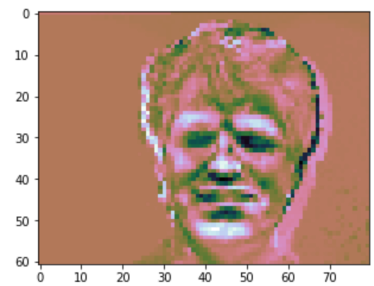
Figure 2: The first three eigenfaces of subject 1.



(a) Eigenface 4

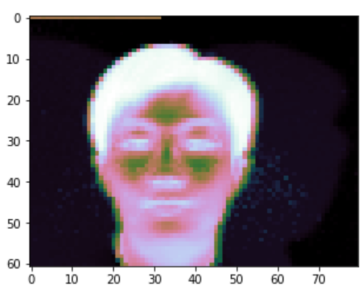


(b) Eigenface 5

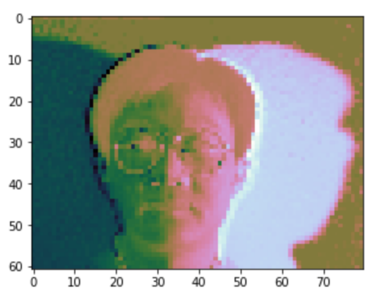


(c) Eigenface 6

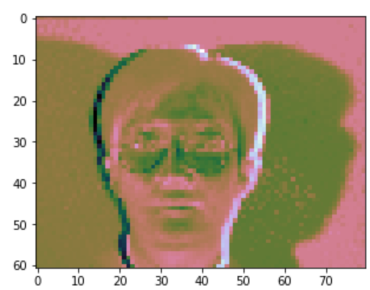
Figure 3: The next three eigenfaces of subject 1.



(a) Eigenface 1

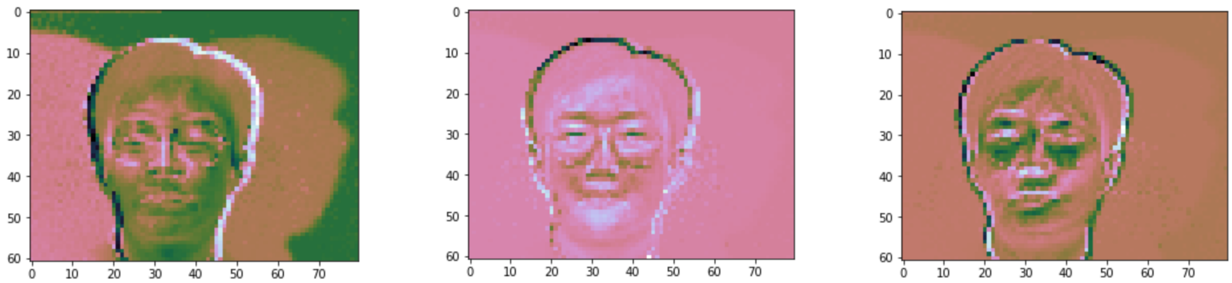


(b) Eigenface 2



(c) Eigenface 3

Figure 4: The first three eigenfaces of subject 2.



(a) Eigenface 4 (b) Eigenface 5 (c) Eigenface 6

Figure 5: The next three eigenfaces of subject 2.

There are several takeaways and patterns that can be extracted by visualizing these top eigenfaces. There seems to be intra-subject and inter-subject similarities and differences. As a quick side note, I chose a different colormap because I found it to be more qualitatively informative than grayscale images. An example of these similarities and differences, looking at the first eigenface (of both subjects), is the fact that high contrast regions seem to focus on the eyes, under-eyes, nose, eyebrows, lips, hair and background. These facial features can effectively be taken as the most imperative for differentiating each subject's images from one another and hence could also be thought of as features to distinguish each subject from another subject. I will come back to this idea when calculating projection residuals. For this first eigenface however, in the two subjects images, the colors/scaling are reversed. For this type of analysis, I find it more informative to look at each end of the scale and/or high contrast regions to signal a feature's importance.

The second eigenface of each subject appears to match almost identically. There is clear importance of the shadowing on each side of the subject's head in the background and little importance of the subject's face. Where there is a difference however is subject 2's face seems to encompass high frequency features such as the apparent wearing of thin-framed glasses which is precisely what is seen in Figure 1(b). Other eigenfaces surely include other even more specific features of importance. For example the 5^{th} and 6^{th} eigenfaces of subject 1 emphasize the tip of the nose and lips whereas the 3^{rd} and 4^{th} eigenfaces of subject 2 emphasize the background and shadowing. As a good sanity check it is also great to see that the eigenfaces from PCA picked up on the biggest difference between the two subjects, the fact that subject 2 has glasses and 1 does not. It is clear that we see the inclusion of glasses-like features in almost all eigenfaces of subject 2 but not in subject 1.

2.2 Face recognition task

Using my implementation of PCA from the previous section I can complete a simple facial recognition task by calculating the projection residuals of the two vectorized test images shown in Figure 1 with the vectorized eigenfaces of subject 1 and 2 respectively. The projection residual is defined in Equation 8 and the projection residuals for each of the combinations of test subject and eigenfaces subject are summarized in Table 1.

$$s_{ij} = ||(\text{test image})_j - (\text{eigenface})_i(\text{eigenface})_i^T(\text{test image})_j||_2^2 \tag{8}$$

i (eigenfaces subject)	j (test subject)	s_{ij} (x10^9)
1	1	2.576
1	2	0.988
2	1	1.039
2	2	2.217

Table 1: Projections residuals of the two test images from Figure 1.

It is quite clear from the results in Table 1 that the projection residual when using the same test and eigenfaces subject is > 2 times more compared to when the test subject differs from the set of eigenfaces.

These results are consistent across both subjects. This means that one could calculate the projection residuals of the test image with the top eigenfaces of several other subjects and determine who the test subject actually is by finding the maximum projection residual and finding the subjects whose eigenfaces it corresponds to. It is fascinating to see that even with just two test subjects and the top k = 6 eigenfaces, there is > 2x discrepancy in the projection residual making for a very clear determination of who the test subject really is.

2.3 Explain if face recognition can work well and discuss how we can improve it possibly.

I believe that facial recognition can work quite well and I think even the simple implementation of PCA for facial recognition in the previous section illustrates that. Solely with PCA (and 6 principal components at that), a simple projection residual formula can easily provide a good face matching. However, I do realize that this task may not be complex enough when there are millions of potential face matches. There is certainly room for improvement using more sophisticate deep learning models like Facebook's DeepFace. DeepFace employs a neural network and is trained on millions of images which ultimately allows it to reach $> 97\%$ accuracy [1] which is most certainly better than what the above implementation would achieve in a similar test setting.

3 Order of faces using ISOMAP

The goal for this problem is to reproduce the ISOMAP algorithm results from the original 2000 ISOMAP paper by implementing ϵ-ISOMAP from scratch. Here, I load 698 face images of different poses of the same face and see if my implementation can successfully qualitatively and quantitatively discriminate and discover patterns of these images in a low dimensional embedding.

3.1 Visualizing nearest neighbor graph

I began by loading each image as a data point and constructing a weighted graph/nearest neighbors graph/adjacency matrix where each entry is the 2-norm distance between data points. For this section I am implementing ϵ-ISOMAP so I only keep entries/distances $\leq \epsilon$. I tuned ϵ to achieve reasonable results and a reasonably sized adjacency matrix. After reading about the dataset, I hypothesized that each image will have roughly two images that are closely related to it. I tuned ϵ such that the adjacency matrix had a number of non-zero entries that represented each data point (or node in the graph) having on average two nearest neighbors (taking into consideration that $\mathrm{A}_{ij} = \mathrm{A}_{ji}$. The adjacency matrix can be seen as a heatmap in Figure 6.

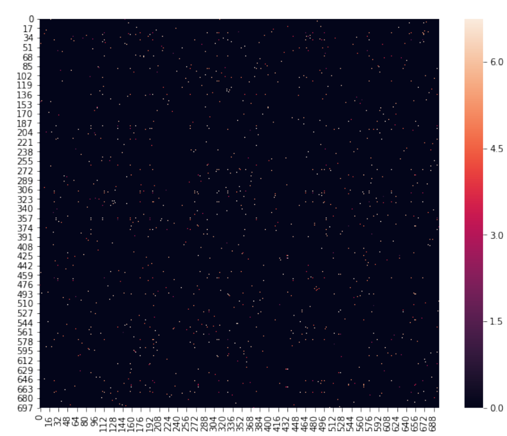
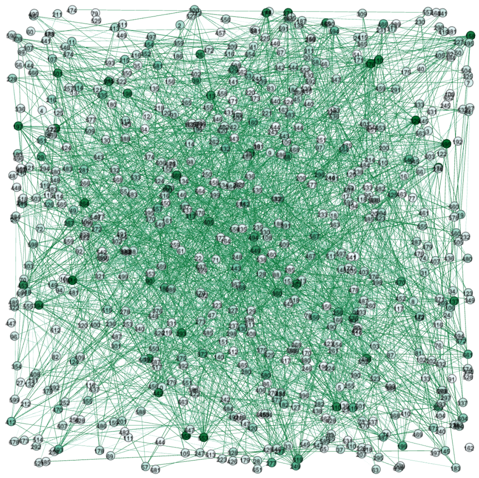
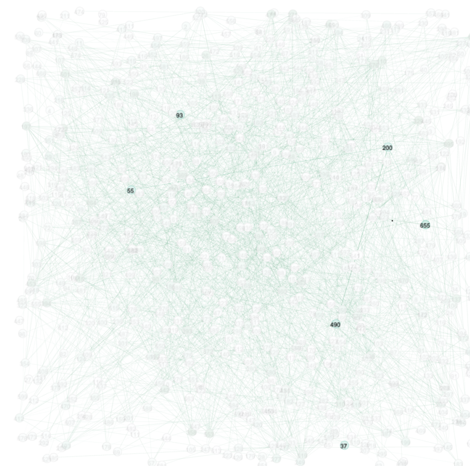


Figure 6: Heat map of the adjacency matrix.

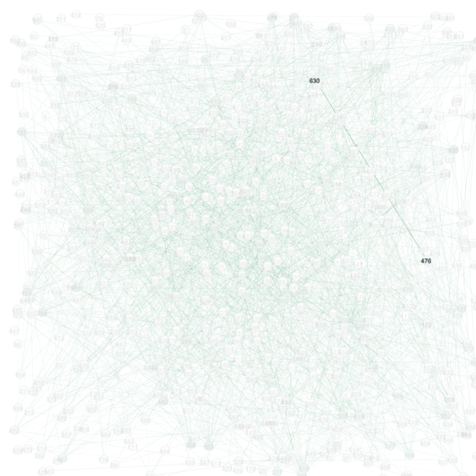
I could better visualize the the nearest neighbors graph using Gephi. Figure 7 shows a visualization of the weighted graph with examples of how some images are more connected to others than some.



(a) Fully connected nearest neighbors graph



(b) Example node with 5 neighbors



(c) Example node with 1 neighbors

Figure 7: Some images have a larger number of similar images than others but 2 on average in this graph.

I am to able visualize these similarities by superimposing some images on top of their corresponding nodes in the graph. Figure 8 shows this informative graph.

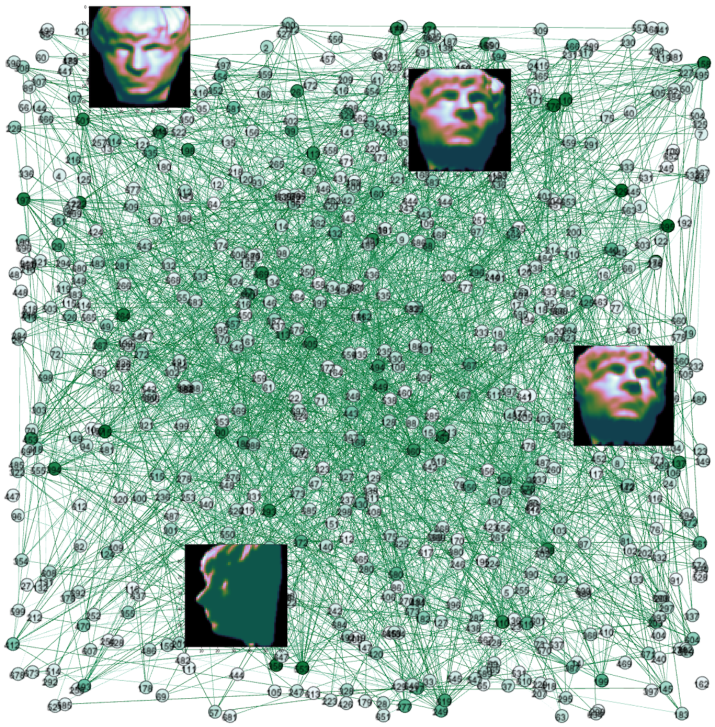


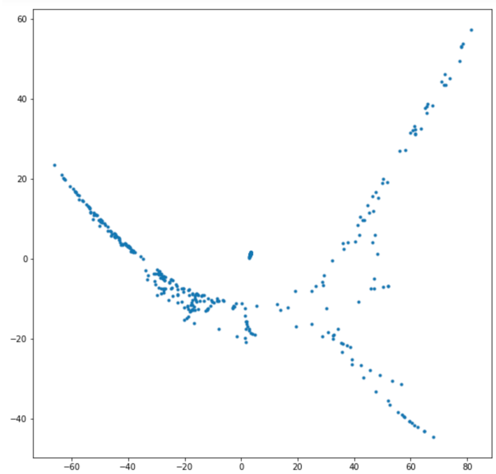
Figure 8: Nearest neighbors graph with face images superimposed.

To prove several arguments, I visualized two images/nodes that are connected to one another, one node that has several connections and is far in the graph (and is not connected to any other image shown), and one node that has one connection and is both far and close to other plotted images in the graph (and is not connected to any other image shown). First, examine the two images on the right side of Figure 8. These are representative images from the two nodes in Figure 7(c). To prove that the connections are working properly, I show that these two images are very similar and nearly the same. Both images are of faces looking up and to the right. The only difference is the one in the upper portion of the graph is looking slightly more towards the center.

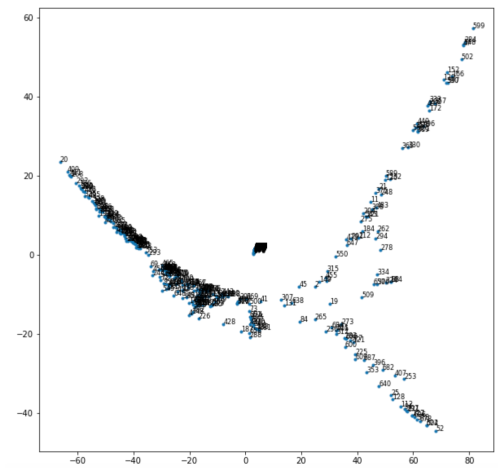
Now, look at the image in the bottom left corner of the graph, this node was highly connected but not to any of the other images shown. This makes perfect sense because the face is looking directly left and likely is very close in 2-norm distance to any other face looking that drastically to the left. Lastly, the face in the upper left part of the graph is not connected to any other image shown which is good because the face is looking towards the center. As a sanity check, although not connected, it is closer in graph-distance to the faces on the right side of the plot than the one on the bottom left which makes sense once again because the face itself looks more like the ones looking up and to the right than directly left.

3.2 Implementing ISOMAP

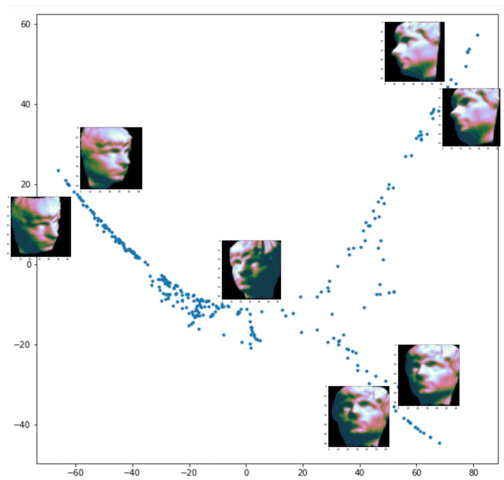
I next implemented the full ϵ-ISOMAP algorithm and obtained a 2-dimensional low-dimensional embedding. I plot the components against one another in the scatter plots shown in Figure 9. Figure 9(a) and 9(b) show the raw plot and the plot with image IDs overlaid so that I could visualize example images from different parts of the plot as seen in Figure 9(c). Figure 9(c) shows a few exemplary images from main areas of the scatter plot.



(a) Raw scatter plot



(b) Scatter plot with image ID overlaid



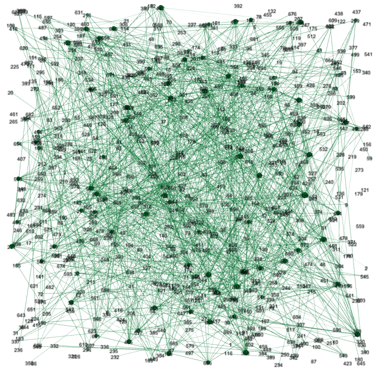
(c) Scatter plot with example images

Figure 9: Images taken from different parts of the scatter plot have different faces and images taken within each portion have similar faces.

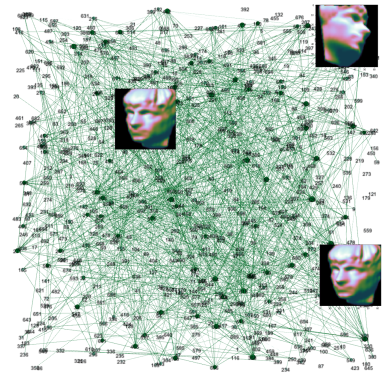
There are several conclusions I can draw from the low dimensional embedding in Figure 9(c). The faces taken from adjacent points in the upper right part of the plot (152 and 502), bottom right part of the plot (25 and 128), middle part of the plot (41), and left part of the plot (20 and 400) all look extremely similar to their adjacent point but different than other faces in other parts of the plot. Additionally, it makes complete sense that faces drawn from the left side of the plot are looking in the opposite direction as faces taken from the right. Also, that a face drawn from the middle is looking more towards the middle. As another way to analyze this plot and as another sanity check, the faces taken from the top right and bottom right portions of the embedding are both looking left but at varying angles which makes sense. These faces from different portions of the right side are still different from one another though which is a great sign.

3.3 Using L1 distance as similarity metric for adjacency matrix

In this section, I aim to repeat the ϵ-ISOMAP algorithm but this time by employing the 1-norm distance as the criterion for being less than ϵ when constructing the nearest neighbors graph/adjacency matrix. Again, I tuned ϵ such that the average number of neighbors for each image was ~ 2. Figure 10(a) shows the nearest neighbors graph and Figure 10(b) shows faces 2, 149, and 437 (top right). Face 2 and 149 are connected and it can be seen that they are in fact very similar faces. Face 437 was not connected but included to prove the inverse and sure enough, the face looks nothing like the other two.

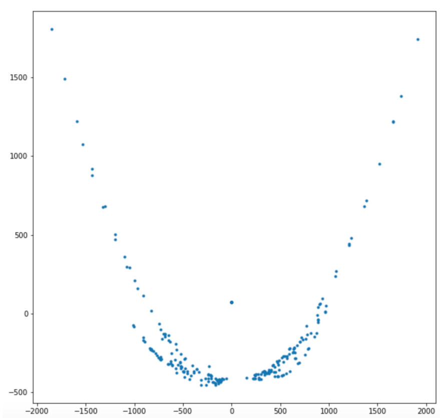


(a) Fully connected nearest neighbors graph using 1-norm

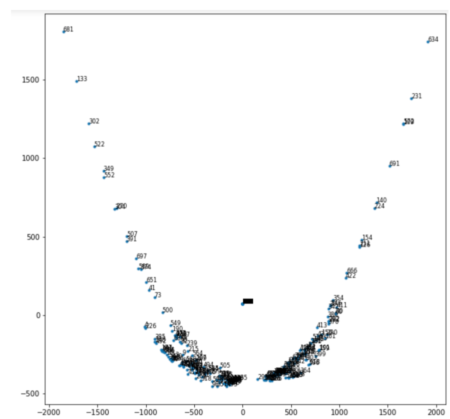


(b) Example images where the top right is not connected to the other two images, which are connected

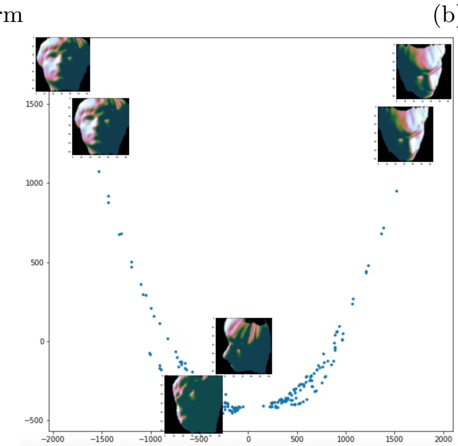
Figure 10: Using L1 norm seems to also work properly as a similarity metric when constructing the nearest neighbors graph.



(a) Raw scatter plot using L1 norm



(b) Scatter plot with image ID overlaid



(c) Scatter plot with example images

Figure 11: Images taken from different parts of the scatter plot have different faces and images taken within each portion have similar faces even when using the 1-norm as the similarity metric.

Figure 11 shows the exact same analysis as completed in Figure 9 but with using the L1 norm. It is quite clear that the embedding is different when using the L1 norm. I can still see clear differences in part of the plot i.e. left branch vs. right branch but an entire branch is lost compared to Figure 9(a). This could signify that when using the L1 norm, some useful information to further discriminate images is lost. On a positive note, the embedding seems to properly differentiate different faces but not quite as well as when using the L2 norm. For example, the faces on either side of the plot in Figure 11(c) are not polar opposites of the other side. However, adjacent points do have very similar faces yet again so the embedding does a reasonably good job.

3.4 PCA on face images

Lastly, I perform PCA on the face images using my implementation in Section 2 to examine whether PCA or ϵ-ISOMAP does a better job at discrimination in the low dimensional embedding. This analysis could also tell whether there are underlying non-linear structures in the data. I computed the eigenfaces for this images to see if it would yield any meaningful results. Figure 12 shows this. As it can be seen, the eigenfaces are nearly a blur which does make sense because of the many rotation variants of the faces but also hints that things become increasingly difficult in PCA when the images acquired are not similar to one another. In this light, using the eigenfaces for meaningful analysis or facial recognition would be a challenging task and may not work. I also plot the first two PCs as a scatter plot in Figure 13. It can clearly be seen that the low dimensional embedding does not do a great job separating the data into more defined sections or clusters as the ϵ-ISOMAP algorithm does. The PCA scatter plot is appealing to look at but is quite homogeneous compared to ϵ-ISOMAP. Now, I can draw the conclusion that ϵ-ISOMAP has a more meaningful projection in the low-d space that can be used to differentiate images. This shows the beauty of the non-linear dimensionality reduction.

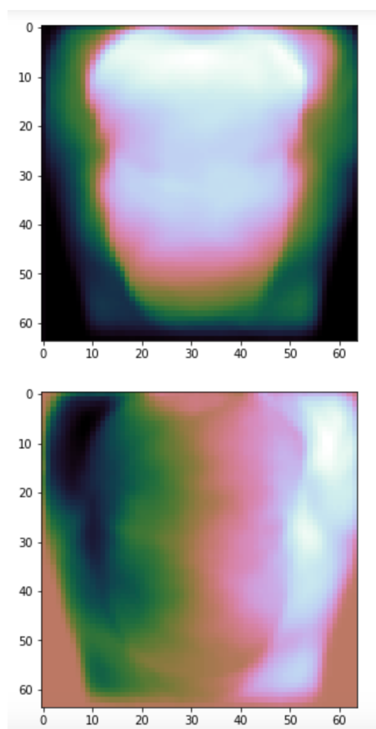


Figure 12: Eigenfaces of faces dataset

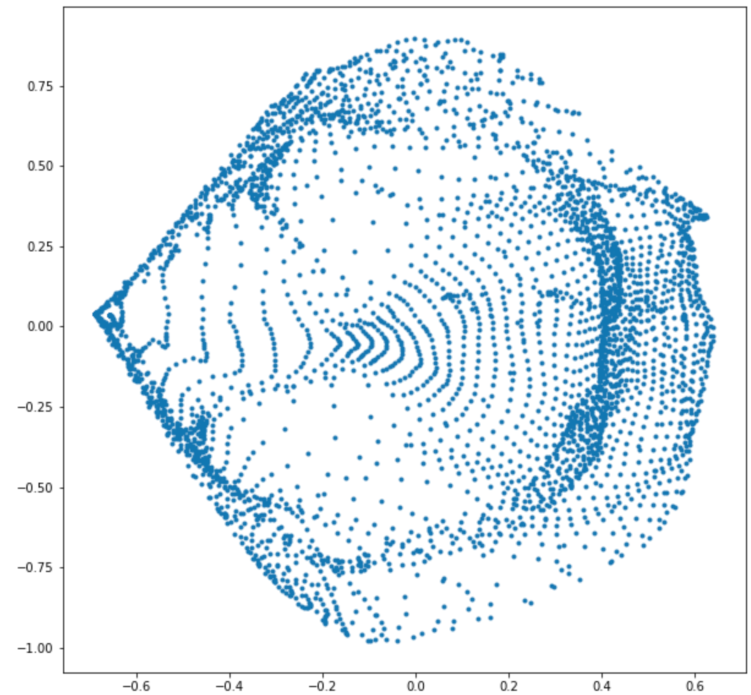


Figure 13: The first two PCs plotted against each other, and hence a low dimensional embedding.

References

[1] https://en.wikipedia.org/wiki/DeepFace