**Question 1**

**What we did:**

In this question, we first loaded the Olivetti faces dataset. It’s a dataset containing 400 images of 64x64 pixels. There are 10 images for each of the 40 unique individuals. We then do a train-val-test split and used a classifier. After that, we reduced the dimensionality with K-Means, do that train-val-test split and used a classifier again. Lastly, we applied DBSCAN to the dataset to group similar images together based on their density.

**Images and/or Rationale:**

Step 1:

**A collage of different faces

Description automatically generated**

The above image just shows what our dataset looks like. There are 40 different people.

Step 2:

A graph of a number of samples for each class

Description automatically generated

A graph of blue and white lines

Description automatically generated

A graph of a number of samples for each classes

Description automatically generated

For the second step, I decided to split the dataset into:

* 80% train
* 10% validation
* 10% test

The images above are from the respective sets and shows the number of samples per person. I choose this split because the dataset only has 400 rows, and that I want more data for training while still being able to validate and test. At first, I thought to use 70-15-15 instead of 80-10-10, but because there are 10 images of 1 person, 80-10-10 allows us to evenly split the samples into 8-1-1 images per person instead of 7-1.5-1.5 images, which is not even. Another choice could be 60-20-20, but I feel like it might not be enough training data.

Step 3:

A close-up of a number

Description automatically generated

The image above shows the result of the prediction on the validation set (97.5%).

Step 4:

A graph with blue dots

Description automatically generated

The image above shows the result of the silhouette score approach. Firstly, technically the “correct” number of clusters is 40 because there are 40 different individuals in the dataset. So, I tried the range from 20 to 100, and I got the optimal K to be 98. However, it seemed like the score can go even higher if K is higher. So, I tried to put the upper bound as 150. The image above is from the second iteration, and that is to try it from 20 to 150. And the optimal seems to be 125. I didn’t try any higher because it seems like the scores start to go down after K is around 140.

As for the similarity measure used to perform the clustering, we’re using Euclidean distance as the measure. I believe that this measure should be appropriate because our data consists of pixel values of images, and these pixel values are numbers. Since they are just numbers, we can effectively use Euclidean distance to measure how close or far 2 images are using those pixel values.

We then reduce the dimensionality to 125 because it was the optimal K value. The image below is again the number of samples for each class after the dimensionality reduction:

A graph of blue lines

Description automatically generated

Keep in mind that the labels of the data are no longer the person id, but it is the cluster id. In other words, the label becomes the cluster that the data belongs to, and there are 125 clusters.

Step 5:



The image above is the result of the prediction on the validation set after dimensionality has been reduced to 125 (85%).

Step 6:

A screenshot of a computer

Description automatically generated

The image above is the result after I ran DBSCAN on the reduced dataset from the previous step (reduced to 125D using K-Means). I tried different numbers of eps and min\_samples, and this seems to be the one that gives the best Silhouette score. However, it is not the perfect parameters because there are a lot of noise points, and the samples aren’t distributed that well between clusters (e.g. cluster 1 has 90 samples while cluster 8 has only 4 samples).

Other than 125D, I also tried to reduce the dimensionality to 2D with PCA just so that I can plot it using the function from the DBSCAN lab from class (F2024\_COMP257\_\_Lab\_2\_b\_dbscan.ipynb). Here are the results:

A screenshot of a graph

Description automatically generated

As we can see, we get a better silhouette score here even though we have less clusters. This might be because the dimensionality is lower, so clustering is easier since there in the higher dimensionality of 125, there could be a lot of overlap and less clear separations.

The similarity measure used here with DBSCAN is Euclidean distance (both 125D and 2D data), the same as Step 4. Again, this similarity measure is used here because our data consists of pixel values, and since they are just numbers, we can use Euclidean distance to measure how similar those pixel values are. For example, images of the same person or images with similar lighting/expression will have similar pixel values.

**Lesson learned:**

* I learned that finding the optimal K for K-Means can take quite a while, and even then, it might not have a high silhouette score.
* I learned that in this case, even though reducing dimensionality can be useful for visualization or faster processing, it can hurt the performance of a classifier (97.5% accuracy became 85%).
* I learned that Euclidean distance is an appropriate similarity measure for facial images due to the nature of images having pixel values, which are numbers we can use to measure similarity.
* I learned more about evaluating clustering results from DBSCAN without plotting any graph. This is because the reduced dataset is 125-D, which we can’t plot, so we must see other things like number of clusters, noise, and how the samples are distributed between clusters.