Hardik Bhardwaj
Eyal Arkin
Project 1 Report

We worked on the project concurrently, bouncing ideas off each other and helping one another understand the concepts necessary to complete the tasks (as opposed to splitting it up). We started by outlining the whole process on paper before implementing it. Then, we started implementing it. For the implementation, we did it step by step, making sure each step was providing the expected results before moving on to the next.

For implementing PCA, we started by loading all the image files as numpy arrays, and we filled an ‘images’ array with all of these pictures. We then calculated the mean of all of the images using the numpy mean function, and then subtracted this mean from all the images and ran the PCA function with K components. For K, we started with 20 and plotted the results. We saw that it was relatively under-fitted, so we tried going up to 200. We then saw that 200 was pretty over-fitted, so we went down to 100 and saw it was still over-fitted. We went down to 50 and it seemed to be good.

In order to display the top 50 eigenvectors, we created a plt subplot with 10 rows and 50 columns and displayed the first 50 eigenfaces we created.

After showing the top 50 eigenvectors, we displayed 5 of the reconstructed eigenfaces alongside their original representations. These comparisons showed that while the eigenface certainly does look different, it is still very recognizable as the same person, indicating that facial recognition should work.

We then fed all 44 test images to the facial recognition model by resizing them to our needs and then again PCA transforming them to the lower dimension. After doing so, we calculate the euclidean distance from the test eigenvectors to the training eigenvectors and pick the one with the smallest value to represent our algorithm’s attempt at performing facial recognition. We decided to test all 44 images to calculate our accuracy score by then going through each comparison made by our code and deciding whether or not it was a match. Our final accuracy score comes out to 90.9% or 40/44.

From this project, the two of us were able to take away the concepts of using PCA on eigenvectors and then using those eigenvectors to create eigenfaces. We were also able to do a deeper dive on the numpy library and expand our understanding of how to implement linear algebra and general math concepts with python.